

# The Engine of Complexity



## EVOLUTION AS COMPUTATION

John E. Mayfield

## The Engine of Complexity



JOHN E. MAYFIELD

---

# The Engine of Complexity

Evolution as Computation

COLUMBIA UNIVERSITY PRESS  NEW YORK

Columbia University Press  
*Publishers Since 1893*  
New York Chichester, West Sussex

cup.columbia.edu  
Copyright © 2013 John E. Mayfield  
All rights reserved

Library of Congress Cataloging-in-Publication Data  
Mayfield, John E.  
The engine of complexity : evolution as computation / John E. Mayfield.  
pages cm

Includes bibliographical references and index.  
ISBN 978-0-231-16304-0 (cloth : alk. paper)—ISBN  
978-0-231-53528-1 (ebook)

1. Biology—Mathematical models. 2. Evolution  
(Biology)—Mathematical models. 3. Biological control  
systems—Mathematical models. 4. Computational complexity. I. Title.  
QH323.5.M389 2013  
570—dc23  
2013005728



Columbia University Press books are printed on permanent and  
durable acid-free paper.

This book is printed on paper with recycled content.  
Printed in the United States of America

c 10 9 8 7 6 5 4 3 2 1

COVER IMAGE: © Getty Images  
COVER DESIGN: Mary Ann Smith

References to Internet websites (URLs) were accurate at the time of writing.  
Neither the author nor Columbia University Press is responsible for URLs that  
may have expired or changed since the manuscript was prepared.

---

# Contents

Preface	xi
Introduction	1
<b>1 The Problem</b>	<b>10</b>
How is Earth different?	10
What is information?	16
What is an evolutionary system?	22
What constitutes scientific understanding?	26
How does computer science fit in?	31
How is purposeful complexity explained by science?	34
<b>2 Computation</b>	<b>39</b>
What is a computation?	39
What does it mean to manipulate information?	45

How is information defined and measured in computer science?	49
Can great complexity be created by short (i.e., simple) programs?	51
Can physical structure be computed?	58
Why have long programs?	66
<b>3 Structure for Free</b>	<b>68</b>
If physical processes compute, what and where are the programs?	68
How do we explain a grain of salt?	72
How does a solar system form?	75
How do nonequilibrium structures form?	78
Must structures have permanent parts?	81
Is behavior a form of structure?	83
Can avalanches be simulated on a computer?	86
Whence structure for free?	88
<b>4 Purposeful Structure</b>	<b>91</b>
What can you get by using instructions that you can't get for free?	91
Can you get a watch without a watchmaker?	97
How do instructions encoded in DNA specify structures more complicated than single proteins?	105
Can instructions dictate activity?	109
What about ongoing activity?	110
So, exactly how do we define purposeful structure?	115
<b>5 Improbability and the Engine of Complexity</b>	<b>119</b>
How do order and disorder relate to each other?	119
How do we reconcile great improbability with existence?	122
Where do instructions come from?	125
How do instructions change our view of object probability?	130
Is life the outcome of a computation?	137

<b>6</b>	<b>Algorithmic Evolution</b>	<b>145</b>
	Can computers learn?	145
	What are evolutionary algorithms?	150
	Can the rules evolve?	161
	How can we visualize what is happening during an evolutionary computation?	164
	Is randomness necessary for learning?	167
<b>7</b>	<b>Evolution Within the Body</b>	<b>170</b>
	How does the human body form from a single cell?	170
	How is the brain wired?	176
	Why isn't the body destroyed by microorganisms?	181
	What does copying add to selection?	189
<b>8</b>	<b>Taking Control of the Cycle</b>	<b>192</b>
	Can the engine of complexity accept any form of information?	192
	Can variation be controlled?	195
	Must selection be natural?	200
	Is there any aspect of the cycle that cannot be controlled?	204
<b>9</b>	<b>Complex Systems</b>	<b>207</b>
	What constitutes a complex system?	207
	How is an electronic device also a network?	209
	What is a biochemical network?	212
	What is a genetic network, and what does it do?	216
	What is a proteome?	221
	What is life?	222
	What is a complex adaptive system?	228
<b>10</b>	<b>Human Learning and Creativity</b>	<b>230</b>
	How are we to understand the human mind?	230
	Where is the information?	231



How are decisions made?	234
Could our world, and therefore our thoughts, be deterministic?	236
What are some general principles of brain function?	239
What is evolutionary epistemology?	241
What are some current theories?	244
How do we invent?	249
What is the evidence that the engine of complexity plays a major role in brain function?	250
What can we say about the origin of human creativity?	255
 <b>11 Cultural Evolution</b>	 256
What is human culture?	256
How do cultures change?	257
How does science work?	261
Economies change, but do they evolve?	265
Do religions evolve?	270
How does the engine of complexity operate in human society?	272
 <b>12 The Evolution of Complexity</b>	 275
Can complexity be defined?	275
How is complexity defined by biologists?	282
How do optimization and coevolution fit in?	292
Do resources matter?	296
How does depth figure in?	299
Is there evidence?	301
 <b>13 Past and Present</b>	 305
How do engines of complexity get started?	305
What does it mean to be in the middle of a computation?	323

<b>14 The Future</b>	<b>325</b>
Where is it all going?	325
Are we about to participate in a new implementation of the engine of complexity?	336
Acknowledgments	339
Notes	341
Glossary	357
References	375
Index	381



---

## Preface

Understanding in a scientific way how complex things, including ourselves, are possible has puzzled the greatest minds for centuries. The very notion of complexity presents conceptual and technical challenges. Many definitions have been proposed and many of these are not quantifiable in ways that allow the property to be measured. I maintain that the concepts of information, complexity, and evolution are so deeply intertwined that only by considering the three together can we make progress toward that long sought understanding. The glue that ties it all together is the concept of computation.

Everyone is familiar with things that are immensely complicated. Living plants and animals present obvious examples as do some products of human inventiveness such as the space shuttle, the U.S. Department of Defense, and supply chains required to produce a modern automobile. Some nonliving, nonhuman things such as galaxies and mountain ranges also seem quite complicated. Are there meaningful ways to compare the complexity of a human being with that of a galactic cloud or the complexity of a coral reef with that of a super computer?

Approaching questions about complexity is never easy. The traditional way, often called complexity science, takes a highly mathematical approach. Many of the practitioners are trained in physics, and many of the examples studied are taken from the nonbiological world, although there are frequent forays into economics and other social systems. I present a different approach, one that is more philosophical and more biological, and one that at first approach is surprisingly nonmathematical. The two approaches complement each other. The nonmathematical route actually explains a number of things that the mathematical route has struggled with and even shows why traditional mathematics is sometimes not helpful. Although this book includes essentially no equations, the approach taken is not motivated by math phobia. The concepts rest firmly on computational theory, a highly technical branch of mathematics. There are no equations, because none are necessary to understand the relevant concepts at a nonexpert level, and because for many readers a mathematical approach would preclude them from reading the book. One of my several goals is bringing to new nontechnical audiences some basic computational concepts that at their roots are very mathematical but whose key ideas can be understood in nonmathematical language.

Richard Dawkins put his finger on a partial solution to the complexity problem in his prizewinning 1986 book *The Blind Watchmaker*, in which he observed that the problem we would really like to solve is not so much the quantification of complexity but rather the genesis of complex design. He noted that some complicated things in our world are prespecified while others are not. Thus, a computer exists because a team of engineers designed it. A crystal-filled rock can also be quite complicated, but it cannot be seen in any scientific sense as pre-designed; it is simply the result of a long sequence of natural forces acting on conditions that were themselves consequences of long sequences of natural processes. In both cases the laws of physics were obeyed, but creation of the computer required something extra: extra data, special information exactly organized in ways not explicitly found in the laws of nature. This is what sets a computer apart from things created in the nonbiological, nonhuman universe, and is what makes it special from a complexity science perspective.

Living organisms also give the appearance of design. Yet, no team of engineers ever planned the details of a jellyfish. Over the last sixty years,

molecular geneticists have worked out a very clear, if still incomplete, sketch of the manner by which a jellyfish is nonetheless prespecified. The key finding, of course, is that jellyfish DNA encodes information that in conjunction with the universal laws of chemistry and physics in the context of jellyfish cells make more jellyfish possible. The requirement for extra, always local, organism-specific information provides a very clear way of contrasting complexities associated with life with complexities generated in other contexts such as rocks, weather patterns, and solar systems. Where, we must ask, does the “extra” local information come from? Meaningful information cannot simply be created out of thin air; it must come from somewhere.

Again, geneticists have come to the rescue by showing how the Darwinian process of natural selection acts on populations of nonidentical individuals. The outcome, over time, is modification of the DNA present in the cells of members of a population of organisms. The DNA of individuals does not change, but over time the DNA in populations does. Modifications to DNA, initially as mutations, and then accumulated by natural selection, constitute incremental bits of information useful for jellyfish surviving in their local environment. Accordingly, random changes in hereditary material (mutations occurring during reproduction) in populations subject to natural selection over a very long period of time establish the origin of the special extra information needed to create each and every living organism. But where does the special extra information come from when a team of engineers designs a new computer or suspension bridge? Is it enough to say that engineers thought it up? I argue not, and will show that when viewed as a computation the same general scheme operating in biology also operates in human cultural and technological spheres as well. This implies that the ultimate source of all the special information that allows prespecification of so many complex things begins as random change subjected to cumulative selection. I call the logical scheme that accomplishes this efficiently the “engine of complexity.” The same engine underlies the evolution of life, social and technological change, and, I hypothesize, human learning and creativity.

*Information* is a trendy word: “We live in a digital [information] age,” “biology is becoming an information science,” “the best hope for unifying general relativity with quantum mechanics is through information theory”—and then there is that “extra information” that

distinguishes the special complexities characterizing so many things on Earth from the ordinary complexity that characterizes a rock, or Mars, or the star Alpha Centauri.

It is clear from our everyday experiences with cell phones, computers, and even our basic ability to respond to what another person says that information can be processed, manipulated, and used to achieve goals. The science that studies this is called computer science. The field is a relative newcomer to the collection of disciplines attempting to explain different aspects of our world. Mathematical analysis of computation places definite limits on what can and cannot be computed. Some interpret the generality of information and its processing to mean that when we observe something changing in the world, we are actually observing the universe computing an outcome. Evolution also can be interpreted and analyzed as a computation. In this book I show how this relatively new approach generalizes the notion of evolution to areas that were previously viewed as evolutionary only in a very loose sense, and show how the evolutionary computation operates in many arenas without direct involvement of DNA.

The computational approach effectively establishes a general theory of evolution and also answers the question of where all the special extra information comes from that is necessary to make all kinds of complex things but which is not encoded in DNA. The surprising answer is randomness: random events and random choices are the ultimate source of all goal-oriented information. This may seem crazy and totally counterintuitive—but then, that is what motivated me to write this book.

## The Engine of Complexity





---

# Introduction

The subject of complexity stimulates lots of questions. To list a few: Why do many complex things seem to have a purpose or to fit to something else in a nontrivial way? Are complex objects different in some fundamental way from simpler objects? How are great complexities even theoretically possible? And how do things that give the appearance of wild improbability actually come about? Answering these and other questions is what motivated me to research and write this book. As I read what others have said and refined my own understanding, I found it impossible to ignore the underlying mechanism that defines evolution. When speaking here of evolution, I mean more than what is typically presented in a biology class, more than the progression of life forms over eons of time. Rather, I am speaking of a common logic shared by a variety of what on the surface may seem unrelated activities. This logic is most naturally seen as a particular computational strategy that when carried out may lead to the creation of complexities that would otherwise not occur.

The computational approach places this work firmly in what some scholars would call the “information school” of philosophical

explanation. This, in my view, brings precision to what otherwise tend to be “fuzzy” concepts and discussions. An example of this is a precise definition of *evolution* that is independent of the life sciences. The definition I provide is applicable to any system where the evolutionary computation is implemented. In this sense, defining evolution in computational terms generalizes the concept without losing precision. Other benefits are the legitimization of the use of computational terms in what on the surface seem to be noncomputational fields, clarification of how thermodynamics applies to evolutionary processes, and clarification of the study of complexity.

Most public libraries have excellent books on biological evolution and others that explain in various ways the physical world we live in. Despite these resources, a recent survey sponsored by the National Science Foundation found that only 45 percent of Americans accept that humans evolved from other animals, and only 33 percent accept the big bang as the origin of the universe. These findings would be easy to understand if we were a culture that rejects scientific technology, but we are not. Perhaps more than any other, over the years, the U.S. populace has embraced the technological fruits of scientific understanding and has benefited economically from this acceptance. Ironically, science in general is held in high esteem while selected aspects of the seamless body of scientific knowledge are rejected for nonscientific reasons. Central to this pattern of denial is rejection of the evolutionary origins of living creatures, especially ourselves. Seeing evolution in computational terms highlights the inconsistency of those who would deny the evolution of life while accepting the same mechanism when demonstrated on a desktop computer and that also explains most of what characterizes human technology. In this book I show it is not just living things that owe their existence to evolutionary computation, but also the ability of your body to fight off infection, the human capacity to learn new concepts, the technologies that characterize modern life, and the various institutions that characterize societies.

In recent years a new dimension of scientific and mathematical understanding has become increasingly important in our everyday lives. This is computer science. The digital revolution is rapidly changing how people everywhere do all kinds of things and how we think about them. In keeping with this, the new perspective is also changing how scientists interpret and understand more traditional fields such as

physics, biology, and engineering, and, not coincidentally, it helps us see how they fit seamlessly together.

Two important concepts linking technology, biology, and computer science are information and evolution. I argue that to properly understand evolution one must tackle the concept of information. This is because the essence of what evolution does is to accumulate information. Surprisingly, the nature of information is a continuing source of dialogue and even confusion among philosophers, physicists, and biologists. The term *information* does not always mean the same thing in different fields or even in the same field, though there are common themes. Interestingly, there seems to be little confusion or controversy in the field of computer science. This is partly because the field could not exist without clear definitions, but possibly also, because computer science may be most fundamental to the nature of reality.

There are at least three things that make the subject of information interesting to me, a biologist, who happens also to be fascinated by larger issues. First, it is obvious to any modern biologist that a proper understanding of life is not possible without a detailed understanding of how the information stored in DNA is utilized to make new living organisms. Second, the process of evolution is very easily understood and illustrated when presented in computational terms. In this mode of thinking, evolution occurs by following a particular strategy for information manipulation and accumulation. In this book I call that strategy “the engine of complexity.” Third, complexity of any significant kind, living or not, is only possible to achieve through processes that can be broadly described as computing. This statement may seem unjustified or even bizarre, but I hope to convince you it is true as you read on.

Life and human enterprise both seem to exhibit higher degrees of creativity than is found in other arenas of the natural world. Expanding on this claim is a major theme of the book; but, suffice it to say that most of us recognize a hedgehog to be more complicated in a creative way than is a granite rock of similar size. Treating evolution as a computation explains in precisely what way the hedgehog is more complicated.

When an evolutionary computation is implemented in a physical system, “creative computations” often produce complex outcomes that would never otherwise be realized. In fact, the link between evolution

and creativity is so tight and so persuasive it is very likely that all sorts of creative complexities we encounter are *only possible* through the agency of evolutionary computation.

The generalization of Darwin's central insights to nonbiological fields, particularly the social sciences, is not at all new; but one of my goals was to find a precise definition of evolution that would allow one to answer the question: "Is process X truly evolutionary (in the Darwinian sense), or is it really something else? The computational definition I present in chapter 5 achieves this goal. Taking an information-centric view of evolution is also not new, but even among my biologist colleagues who are successful scientists and fully accept the Darwinian explanation of life there are many who do not appreciate the power and reach of the evolutionary computation outside the immediate domain of biology.

The origin of complexity is a puzzle that engaged Plato and Aristotle 2,500 years ago and has fascinated most leading philosophers ever since.<sup>1</sup> The most active current arena for this ongoing discussion is cosmology, where physical scientists are probing fundamental questions concerning the origin and nature of our universe. A major thread of the current discussion attempts to integrate principles into physics that are provably true in theoretical computer science. To quote Seth Lloyd, a theoretical physicist at MIT, "we can prove mathematically that a universe that computes must, with high probability, give rise to a stream of ever more complex structures."<sup>2</sup> Perhaps even more compelling is the argument that no complex result can ever be achieved except through computation. A computation is defined in this context as any process that manipulates information.

My personal intellectual odyssey, of which this book is the latest expression, began twenty years ago in 1993 when I first read Richard Dawkins's *The Blind Watchmaker*.<sup>3</sup> I was teaching introductory biology to college freshman at the time. The text I was using provided an adequate introduction to evolution, and I had little trouble presenting and explaining the standard scientific narrative to the apparent satisfaction of most students who had not yet thought much about the subject. But, I had a nagging feeling that the standard way evolution was being taught lacked something important.

I read Dawkins in search of enrichment for students who wanted to learn more; and, frankly, I hoped to learn more myself. In college I majored in physics and my graduate education was in biophysics.

I did not receive formal educational training in evolution. In spite of this, the fact of biological evolution was never an issue. For most of my career I studied DNA sequences, and anyone who does this faces the obviousness of evolution every day. Reading Dawkins reemphasized for me the centrality of “information” to the concept of evolution. At one point I even wrote in the book margin, “it’s all about information.” At the time I didn’t grasp many of the implications.

The story turns out to be more remarkable than I ever imagined. Understanding how information and evolution are intertwined explains many things that on the surface don’t show obvious connections. It is now clear to me that most things we care about in our everyday lives are best explained in terms of evolving information. The power of the evolutionary strategy when combined with information use is so great and so pervasive that I have come to believe teaching how complex things come about should be a cornerstone of our entire education system. Sadly, even in 2013, it is not easy to find a single course in most university curricula focusing on the interdependence of these two concepts. While this book is written to engage a broad range of academic, student, and well-informed lay readers, it is also my hope that it will serve as a foundation for new creative courses probing the significance of information, computation, and evolution to the human situation.

The synthesis I offer is one that any educated person could piece together by reading extensively in multiple sources. It is helpful, however, to have a guide. Part of the problem is that to understand the dependence of evolution on information, and vice versa, one needs to view the world in a way that many noncomputer scientists find foreign. Another challenge of self-study is dead ends. In one false start I spent an entire year exploring and understanding the thermodynamics of evolution only to conclude that that line of inquiry yielded little useful insight. The key for me was coming to realize that the process of *evolution is most meaningfully understood as computation*. Once I made this intellectual leap, it was clear that understanding the thermodynamics of evolution was no more difficult than understanding why a desktop computer only works when plugged in. The previously mystical nature of the thermodynamics of evolution dissipated. Another false start was my search for a “law of evolution.” Surely, I thought, evolutionary processes exhibit some regular property that can be captured in the form of a simple law. Such laws have been sought by a lot of very smart people

for the past 150 years. The best that these efforts have accomplished is to separate the twin pillars of Darwinism, diversity generation and selection, and to formulate “laws” of increasing diversity and of natural selection.<sup>4</sup> This to me is disappointing. Seeing evolution in computational terms helps explain why laws of evolution are so elusive.<sup>5</sup>

I have found that a fruitful, if nontraditional, way to approach the notion of complexity is to distinguish simple undirected complexities like rocks and galaxies—things that form spontaneously according to the simple rules of chemistry and physics—from intricate purposeful complexities such as exhibited by oak trees and jet airplanes. These things only come about because of the prior existence of complicated rules; think of them as “instructions.” In this book I argue that the information inherent in all but the most trivial instructions always originates in an evolutionary computation. The implications of this claim are profound.

When writing early drafts of these chapters, I sometimes found myself imagining that my arguments would be so powerful, so persuasive, that someone who previously rejected the evolutionary explanation of life for religious or philosophical reasons would be forced to undergo an intellectual conversion and embrace a deep scientific truth through logic alone. I now accept that this is not likely to happen, for logic is rarely the impediment. The logic of Darwinism has been clear for 150 years. But, for those who reject it, a more general way of thinking about evolution—one that explains life, the immune system, human thought, science and technology, and modern economies in a single unified way—probably won’t be convincing either. Often a person’s reason for rejecting the explanatory power and creativity of the evolutionary process is not based on scientific logic but on fear. For those who fear that deep understanding might diminish their faith and lead to despair rather than hope, I offer that a universe of infinite creativity is not such a depressing place after all.

Other authors have presented the idea that the evolution of life employs a simple information-processing strategy. Daniel Dennett calls the strategy “the evolutionary algorithm.”<sup>6</sup> Richard Dawkins emphasizes the essential role of “replicators,” entities or agents that make copies of themselves. I focus rather on the critical importance of information being organized into the form of instructions and show that the evolutionary process of replication, modification, and selection

(Dennett's algorithm using Dawkins's replicators) is the only practical way to create this form of organized information.

The perspective I present provides a way of understanding our world that has only recently been woven together by thinkers in specialized fields ranging from physics to philosophy but is not generally recognized as a formal field in its own right. The synthesis relies on deep principles of computer science but does not require mastery of computer theory to appreciate. I present no formal mathematics because none is needed to understand the central arguments and because the relevant mathematics is so abstract that presenting it would rule out most of my target readers.

While developing these ideas, it was surprising for me to learn that the humanistic concepts of purpose and creativity can be explained and understood in computational terms. We will see that purpose emerges naturally from repeated selection and that creative outcomes are a natural consequence in many evolutionary systems. I argue that, fundamentally, all creativity stems from the inexhaustible well of "randomness," or more precisely, random choice. The process of evolution is so effective at tapping into this source that when properly employed it makes almost anything possible. Who would ever imagine octopuses or desktop computers if they had not personally encountered them?

From time to time the accepted scientific perception of the most informative way to view our world undergoes a fundamental change—or paradigm shift.<sup>7</sup> Because science and society are so closely interconnected, these changes profoundly influence how society at large views itself and its situation. I, and others, see science and society to be in the midst of such a transition. Most scientific principles discovered over the past three or four hundred years have been expressed as mathematical formulae that describe regular behaviors of specific aspects of the universe. Isaac Newton's law of gravity is a good example. He attributed to all objects a property called mass and then hypothesized that every mass attracts every other mass by a force called gravity. The law of gravity states that the strength of this force is proportional to the product of the two attracting masses divided by the square of the distance between them. In the grand scheme of things, this is a simple relationship. Since Newton's time (he died in 1727), scientists have discovered a large number of relatively simple relationships that characterize regularities observed in the physical world.



In reviewing the enormous achievements made possible by the discovery of mathematical relationships, it is tempting to generalize and conclude that everything can be described just by discovering the appropriate equations. One consequence of the equation-based view of the universe is its resulting “determinism.” Simple equations are rigid and unforgiving: fill in the same numbers and the results are always the same. But life is not like that; neither is human society. Simple relationships do not explain everything. Many phenomena are too complex to be described by simple equations. There is no equation that can substitute for a computer-based word processor, and if the history of life were to be rerun, there is no reason to believe the results would be at all the same. Quite the contrary, evolutionary theory makes clear that if the origin and progression of life could be repeated over and over, each time the details would differ.<sup>8</sup> Similarly, human activities are unpredictable in contrast to the predictability we have come to expect in the physical sciences. This suggests the need for new ways of seeing and exploring our world.

Increasingly, scientists in many fields are suggesting this new way is in terms of information. We live in an information age. Everywhere things are digital: cell phones, TV, movies, clocks, stock markets, you name it. More and more, activities are made faster and more reliable by computers. Newer cars require them, airplanes cannot fly without them, and smart phones and household appliances use microprocessors. This transformation is not just occurring in gadgetry but also in the foundations of scientific understanding. Quantum mechanics is being reinterpreted in informational terms. In 1990 John Wheeler of Princeton University, one of the twentieth century’s most respected theoretical physicists, coined the phrase “it from bit” to make the point that the very foundations of our physical universe are informational. He hoped the phrase would challenge younger theoreticians to rethink how the universe might be understood at the deepest levels. Seth Lloyd, professor of mechanical engineering at the Massachusetts Institute of Technology, believes this line of thinking has the best chance of solving the greatest current challenge in physics—unification of the theory of general relativity with quantum mechanics.<sup>9</sup> Biology too is increasingly seen as an information science. DNA was recognized more than fifty years ago to store organism-specific information in its chemical structure. Much of current biological research focuses on how this

information is used to make and maintain living organisms. Psychologists increasingly accept the primary function of the human brain to be information processing.<sup>10</sup> Social institutions and interactions are also being explained ever more frequently in informational and evolutionary terms.<sup>11</sup>

The single word that encompasses information processing is *computation*. Any manipulation of information can be treated formally as a computation, and the scientific field that studies limitations and opportunities afforded by the manipulation of information is computer science. As a formal field of study, computer science began in the 1940s as part of the wartime effort to build computers to break codes and calculate artillery shell trajectories. Since then the field has undergone enormous change both theoretically and practically.

From physics to the social sciences it seems that every field of study is being reevaluated in terms of information and its manipulation. The reason is simply that an information-based perspective often allows a deeper level of understanding and provides solutions to otherwise intractable issues. This book explores a few examples. Because I am a biologist, many of my examples come from the life sciences, yet I believe the book belongs as much or even more in the domains of computer science and philosophy.

The book is organized so that each subchapter heading is a question. This mirrors the nature of science, which fundamentally is about asking and answering questions. This device focused my writing and, hopefully, also will help keep readers on track.

Finally, a disclaimer is in order. When introducing technical, particularly mathematical, concepts to nontechnical audiences, it is necessary to simplify, to extract the essence of a concept without getting bogged down in technical detail. Typically, a complete technically correct definition is incomprehensible without extensive study and background. Because of this, I am very aware that experts may take issue with some of my explanations. In such situations they should ask, is the simplification presented so misleading that the reader would be better off with no explanation at all? Or does it simply lack detail and nuance?

# 1

---

## The Problem

### How is Earth different?

Imagine you are an intergalactic scientist observing our world. Your instruments tell you the atmosphere is wildly out of chemical equilibrium with too much oxygen and too little carbon dioxide. Sister planet Venus, in contrast, has 20,000 times more carbon dioxide and no free oxygen in its atmosphere. Earth also emits complex radio signals at many different wavelengths, and parts of its surface give off light at night even though the surface is not hot enough to cause rocks to glow. A closer inspection reveals narrow strips of concrete thousands of miles long, skyscrapers, and ecosystems dominated by rows of single plant species. Still closer inspection reveals trillions of freely functioning organisms. Many, called plants, individually carry out incredibly complex chemical reactions using the energy of sunlight to create chemical and physical structure releasing oxygen. Others, called animals, eat plants or eat animals that eat plants, breathe in oxygen, and carry out similarly complex interlocking chemical reactions that convert plant chemical structure into animal chemical structure and temporarily

sequester the chemical free energy required for independent activity. And, there is more. One very successful animal type generates intricately structured nonliving objects that greatly enhance its physical and mental capabilities. This species modifies the natural environment for its own benefit. The roads and skyscrapers observed from afar are two examples of large numbers of such environmental modifications. The species also exhibits complex modes of communication and has created intricate social institutions. An outside observer capable of interstellar travel would instantly recognize this riot of complexity to be the telltale sign of a planet that harbors evolutionary systems. Why would she draw this conclusion? Answering that question is what this book is about.

Humankind has sent rocket probes to Venus, Mars, Jupiter, and Saturn's moon Titan and studied the planets and their moons with ground- and space-based instruments. All the planets except Mercury have atmospheres; so do all the larger moons except our own. The surfaces we observe have mountains and craters unless they are too soft to support such structures. Several moons are covered with deep layers of ice and one, Io, a moon of Jupiter, has sulfur volcanoes. Scientists have identified features both familiar and strange, but everything discovered is interpretable as the natural outcome of known chemistry and physics applied to conditions that resulted from those same laws acting on earlier conditions. The chain of explicable events extends back to the beginning of time, and never do the explanations require conditions whose likelihood defies credibility. Improbable things happen and have happened throughout history. Science has not yet explained everything, but there is no persuasive evidence for the occurrence of miracles, of events so improbable that there is no reasonable expectation of them happening somewhere, sometime, in the lifetime of the universe.

Earth is different from the other planets we have studied. While we know of nothing in our world that violates the laws of science, textbook chemistry and physics do not fully account for iPods, blades of grass, and Beethoven symphonies. The laws of chemistry and physics permit them but do not predict them, and the probabilities of these things occurring by random comings together seem inconceivably small—yet they occur in abundance.

Evidently the presence of life changes the nature of the probability calculation, but how? What bestows life with the ability to generate

seemingly endless streams of features that no person would predict if he knew only chemistry and physics? What, if anything, about the presence of life also suggests skyscrapers and computers? What principle is shared by photosynthesis and the Sears Tower in Chicago that does not operate on Mars? What is it about some things, particularly products of human ingenuity, that lead us to see them as somehow different from the products of nonbiological nature?

To fruitfully explore these questions, we need to develop a particular viewpoint. We need first to see things in terms of parts and subparts. This is hardly new. The Greek atomist school of philosophers, in the fifth century B.C.E., argued that the whole of reality consists of small particles—atoms, in empty space—and that physical objects represent instances of successful random comings together of atoms. Science and philosophy have come a long way since then, but aspects of the idea still resonate.

The modern view is that physical objects are made of atoms and molecules. Random assembly of these parts would certainly create something different almost every time. Given enough time and appropriate parts, random assembly should be capable of producing any conceivable object. In this sense, randomness encompasses all possibilities. The dilemma, recognized long ago by Plato and Aristotle, is that unfettered randomness rarely results in anything of interest. If the number of parts is large, then any one configuration is naturally viewed as very improbable in comparison to all the apparent alternatives. The number of unfruitful ways of putting things together is so large that you might have to wait through a million trillion lifetimes of the universe before a particular structure spontaneously appeared.

The objects we care about most tend to have complicated histories and exhibit intricate and highly specific features. Something else calling out for explanation is the distinct impression that earthly structures have purposes. They seem designed for something, or to fit something, or at least to “make sense” in some context. This raises another question: Is there a scientific explanation of purpose? It turns out there is: evolutionary processes naturally create qualities we perceive as purposeful.

An important dimension of the scientific argument for evolution sometimes overlooked by detractors of biological evolution is that the conceptual problem posed by earthly complexity is not limited to

living organisms. From symphonies to space shuttles our lives are filled with nonliving complexities at best awkwardly explained by the laws of physical science and simple probability. Living organisms and most products of human enterprise require more than Newton's laws and quantum mechanics if they are to be fully understood. This something extra is the concept that ***information when properly organized and employed can instruct the formation of very specific and otherwise highly improbable structures.*** Conscious exploitation of this principle has made possible all the technological advances that characterize modern civilization.

A related principle is: ***The apparent improbability of life-associated and technological complexities mirrors the amount of the extra information required for their formation.*** Every living thing and many non-living products of human ingenuity illustrate this. The more information required for something to be formed, the more improbable it seems. Contrast, for example, a stone axe with a supercomputer, or a simple virus with the human being. The more complex, and hence improbable, something seems, the greater the amount of specialized information needed to produce it. The form of this specialized information is variously recognized as blueprints, recipes, genes (i.e., DNA or RNA), or more generally as *instructions*. Instructions come in many forms, but a simple definition might be as follows: ***"Specific encoded information having the property that when used to sequentially modify a second system, the final outcome is predetermined and somehow advantageous or desirable to whoever or whatever created the instructions."*** Instructions embody information that is goal oriented. They never get created unless the outcome benefits the creator. Generally instructions specify or predetermine specific steps or actions to be carried out in a second system. This second system may be, but is not always, a human being. Completion of the steps always results in the achievement of a specific goal. Because of this, a goal is implicit in every instruction.

When followed, instructions often lead to the formation of objects or actions that otherwise would be quite impossible to achieve because of their absurdly low probabilities; but with appropriate instructions and systems to implement them, the resulting structures can be understood as not improbable at all. It is clear that to understand life and the creations of human ingenuity we must explain the origin of instructions. The definition of instructions given before is rather abstract and

likely to be unsatisfying to most readers at this point in the book. An example will help bring it to life. Others occur later.

The basic idea of an instruction is well known to everyone and is well illustrated by a kitchen recipe. A soufflé is a food dish that doesn't come about without human activity. To make a soufflé requires that a series of exacting but simple steps be followed; and when these steps are followed, the result is special. A recipe for a cheese soufflé taken off the Internet<sup>1</sup> is given here:

1. Separate 6 eggs into yolks and whites.
2. Melt 1/4 cup of butter and blend in 1/4 cup of flour, 1 tsp of salt, 1/4 tsp of ground mustard, and a dash of cayenne. Heat and stir for 2 minutes.
3. Slowly add 1 1/2 cups of milk, whisking constantly to make a smooth white sauce. Heat 2 more minutes, stirring constantly, then remove from heat and stir in 1 cup of grated cheddar cheese.
4. Whisk the egg yolks to break them up, add a few tablespoons of the hot cheese sauce to warm the egg yolks, and then add the warmed yolks into the cheese sauce and whisk the mixture.
5. Beat the egg whites with an electric mixer until shiny and floppy white peaks are formed when the beater is withdrawn. Whisk 1/3 of the whites into the cheese sauce. When well mixed, fold the remaining 2/3 of whites into the sauce.
6. Pour the mixture into a 2-quart flat baking dish or pan, and cook at 400 degrees (Fahrenheit) in a preheated oven for 12–15 minutes.

This recipe, if followed carefully, will result in a delicious fluffy dish that contrasts greatly from what you would get by simply mixing the same ingredients together in a frying pan and cooking them over a fire. A soufflé is more complex (and harder to make) than scrambled eggs. Because of this, it is special. How special? I argue that its specialness is measured by the complexity of the recipe. In a sense, the amount a soufflé is more complex than scrambled eggs is measured by the greater number of steps required to make it. In this example, the increased complexity is not huge; we often encounter far greater complexities. Even so, it is unlikely that any but the most creative chefs would ever make a soufflé without following a recipe or at least having watched someone else make one. This particular instruction is one-dimensional

and linear, others may specify parallel (simultaneous) steps, and yet others such as blueprints are two-dimensional. Whatever their form, they encode specific goal-oriented information concerning actions to be taken by someone or something.

But not everything requires instructions. To help make sense of the seeming infinitude of objects and systems of objects and behaviors of objects that populate our environment, it is useful to divide everything into two categories. One category includes all those things whose formation requires instructions (or recipes or blueprints); that is, they require extra information not explicitly found in the universal laws of chemistry and physics. Let us for convenience call things requiring instructions for their formation *Type II objects*. This group includes all living organisms, and most products of human intellect from soufflés to spaceships. The other category includes all things not requiring instructions. Let's call them *Type I objects*. This group includes most things that are not directly a product of life or human activity such as rocks, oceans, weather patterns, and solar systems. Taken together, the two categories encompass everything.

Dividing everything in this way focuses attention on the extra information some, but not all, things require for their formation. It also answers the question we are working on. *Type II objects* are what make Earth different. All known Type II things are either found on Earth or were made on Earth, and Type II structures distinguish Earth from the other known planets. It is possible, of course, that there are unknown planets that exhibit Type II structures. Earth also exhibits lots of Type I structures, but this does not set it apart from other astronomical bodies in the solar system and elsewhere. The instructions that characterize the formation of Type II objects encode purposeful information. Instructions always have an intended outcome. They are inherently purposeful. Thus, three tasks before us are to learn what is meant by "information," to understand the origin of instructions, and to discover where the purpose inherent in instructions comes from. It turns out that evolutionary computation nicely explains the origin of instructions and of purpose.

To recap this section, when viewed from the narrow perspective of chemistry and physics, many things in our world are so complex and so improbable that it would be easy to conclude that their existence is impossible without the intervention of an intelligent



power beyond nature. The notion of instructions provides a non-miraculous alternative explanation.

## What is information?

We hear all the time that we live in the information age, and if you are like me, you are constantly amazed by its uses and its consequences. Cell phones, televisions, and computers are “digital.” Their functions are based on information encoded in sequences of 0s and 1s. These sequences encode information. Somehow patterns of 0s and 1s can be used by a suitable device to produce something we want. Look in your favorite dictionary and you are likely to find a definition of *information* that relates information to acquired knowledge. There is an expectation that information, whatever it is, must be useful or have meaning to someone or something. Peter Denning and Craig Martell, as part of the Great Principles of Computing<sup>2</sup> project, define it thus: “Information is meaning conveyed by patterns of symbols.” On the surface this seems pretty straightforward and so does another important property of information: that it can be encoded into messages that can be stored and transmitted via various media. In fields other than computer science, scholars often equate information with acquired knowledge.

Mathematical definitions of information are possible and useful but fall short of these definitions. Mathematical definitions allow us to calculate the amount of the stuff (information) that is present in a particular occurrence, such as a particular sequence of symbols, but cannot tell us how much meaning it has, or indeed, whether a sequence of symbols is useful for anything. This harsh assessment is modified somewhat if we know how the information in question was generated. With evolutionary systems and more generally with any system that learns, we know any information acquired must capture some aspect of whatever is being fit to (evolutionary system) or learned about. This observation does allow meaning to be loosely quantified under the assumption that the information wouldn't be there if it didn't mean something.<sup>3</sup>

Since quantitative definitions underlie the information revolution, it is important to understand what mathematicians and engineers mean when they speak of information. Let's begin with a little history. In the late nineteenth century scientists began investigating deep implications

of a particulate world made of atoms and molecules and relating this new perspective to the already established field of thermodynamics. Classical thermodynamics is based on “systems”: collections of matter with boundaries. Systems can be complex machines, or chemicals reacting in a beaker, or even a room full of air—pretty much anything with a boundary. All systems are characterized by energy, entropy, and matter. Thermodynamics is the science of how energy and entropy change as systems change. The first law of thermodynamics states that energy can change form but never be created or destroyed. Some changes, but not all, allow work to be done. In thermodynamics, *work* is a general term that means to perform an activity involving energy change. Activities like lifting a box, driving a car, or creating new chemical structure involve work. In the process of work, energy is conserved but changes into a form that is less able to do more work. *Entropy* is the term used to characterize the capacity of a system to do work. The *higher* the entropy of a system, the *less work* it can do. The second law of thermodynamics states that the entropy of an *isolated system* must increase or remain the same; it can never decrease. This well-established rule can be very misleading because few systems in our everyday experience are isolated. When the entropy of an isolated system reaches a maximum, the system is said to be at equilibrium and no further work can be done. In general, a system can change until it reaches equilibrium, and then it cannot change again unless some outside influence acts on it (work is done on the system or something is added to or removed from it).

A moving automobile is an example of a system proceeding toward equilibrium. If you rigged your car so that the air required by the engine was provided from a pressure tank and you captured all the exhaust and heat produced by the engine in an insulated bag, you would have a rough approximation of an isolated system. The modified car would drive just fine until it ran out of gasoline (or air). Then, it wouldn't move; it would be at equilibrium. The total amount of energy in the system wouldn't change as you drove, but gasoline and oxygen (in the air tank) would gradually be converted to carbon dioxide, water, and heat in the bag. When the gasoline (or oxygen) was all converted, the car wouldn't move again until you added more gasoline and replenished air in the pressure tank. Putting the carbon dioxide, water, and heat back in the gas and air tanks wouldn't help; the car requires a low entropy form of energy provided by gasoline and molecular

oxygen (in air). Once converted to carbon dioxide, water, and heat, the amount of energy in the system would be the same, but the quality of the energy present would be degraded to the point where no further work to move the car could be done. It is possible to convert the carbon dioxide and water back into gasoline and molecular oxygen, but this requires the expenditure of low entropy energy from an outside source.

German physicist Ludwig Boltzmann and American chemist Willard Gibbs laid the conceptual and mathematical groundwork for a molecular understanding of entropy. In this conception, a physical system consists of a large number of atoms or molecules that at any particular instant are each characterized by a position and velocity. Any instantaneous configuration of all the molecules in a system is called a microstate. Because molecules are in constant motion, systems are constantly transitioning from one microstate to another. An observable state (something we can see or measure) is actually a time average of the many microstates occurring during the period of measurement. With this conception, entropy measures the total freedom of a system to explore microstates; the more microstates available to be spontaneously explored, the higher is the entropy. This also means that entropy measures uncertainty. The higher the entropy, the more uncertain one is of the instantaneous microstate a system is actually in. Systems that are somehow prevented from accessing some microstates have structure (and lower entropy than if there were no structure).

In the 1940s and 1950s Claude Shannon adapted the physical concept of entropy to meet a need of the emerging communications industry. The problem was how to engineer a system so that it could transmit a desired amount of information in a prescribed time. To solve this problem, it was necessary to devise a quantitative measure of the capacity of an information-transmitting system. Shannon's basic idea had two parts: a system has the potential to carry information if it can take on multiple states or configurations; and, the more possible configurations, the greater the information embodied by any one of them. The measure Shannon settled on is based on probabilities. Because the probabilities of all possible alternatives available in a particular situation must add up to one (one thing actually happens), the more alternatives, the lower the average probability of any particular one occurring.

To give an example, three letters comprise the word *the*, and there are twenty-seven different ways of making three-letter words from the letters *t*, *h*, and *e* if the letters can be used more than once (*thh*, *the*, and *eee* are three of the twenty-seven). The probability of the first letter being a *t* is  $1/3$  (it must be either *t*, *h*, or *e*). Likewise the probability of the second letter being an *h* is also  $1/3$ . According to the rules of probability, when an event is made up of sub-events, each with its own probability, the probability of the combined events is calculated by multiplying the probabilities of the sub-events. Thus the probability of the word *the* being formed by sequentially drawing letters from an envelope containing equal numbers of the letters *t*, *h*, and *e* is  $1/3 \times 1/3 \times 1/3$ , or  $1/27$ . Now consider the word *evolution*. It is nine letters long and uses eight symbols. The number of different ways to arrange nine letters using an alphabet of eight letters is  $8 \times 8 \times 8 \times 8 \times 8 \times 8 \times 8 \times 8 \times 8$  ( $8^9$  in scientific notation). This turns out to be 134,217,728. The corresponding probability of spelling out the word *evolution* by sequentially drawing nine letters from a box containing many scraps of paper each with one of the letters *e*, *v*, *o*, *l*, *u*, *t*, *i*, or *n* written on it is  $1/134,217,728$ . If you should decide to try this, plan to devote a long time to the project! The number of possible sequences increases very rapidly as the length of the sequence increases and also as the number of permitted symbols increases. Thus the probability of forming the word *evolution* by randomly drawing from the full twenty-six-letter English language alphabet instead of just the eight letters actually used is  $1/26 \times 1/26 \times 1/26 \times 1/26 \times 1/26 \times 1/26 \times 1/26 \times 1/26 \times 1/26$ , approximately one divided by 5 trillion (12 zeros). The words in this sentence constitute a specific configuration of 118 letters, spaces, numbers, and punctuation marks. There are approximately  $1$ , followed by 176 zeros, configurations of 26 letters, 2 punctuation marks, 2 numbers, and one space (31 total symbols) 118 symbols long. This is a very large number of alternatives. It dwarfs the number of atoms in the visible universe, and it is only one sentence! Imagine the number of alternative books.

Communication requires the discrimination of one configuration or sequence from possible alternatives. From the perspective of a communications engineer, the larger the number of alternatives that can be discriminated among, the more information that can be carried by a specific transmission. Writing is so useful for communicating because each document is very special—there are a lot of alternatives. A fairly

short message in English carries a lot of information. But remember, the engineering perspective on information cares not a whit about the meaning of the message, only how special it is.

The number of alternatives, the probability, the length of a message, and the amount of information a message carries are all closely related. Long messages can carry more information than short ones, and long messages are vastly more improbable. Shannon needed an information measure that was additive; in other words, if two messages are characterized by information  $X$  and  $Y$ , respectively, the combined information should be  $X + Y$ . Length has this property, but probability does not. It turns out that this was essentially the same mathematical problem faced by Boltzmann. The difference is that in information science systems consist of sequences of symbols, while in physics they are made up of moving molecules in space. Believe it or not, these seemingly very different situations can be mathematically framed in a similar way. Shannon's genius was to recognize this.

The measure Shannon proposed for the information content of a message involves taking the probability of each allowed configuration (each ordering of the allowed symbols permitted by the system), dividing it by the logarithm of that same probability, calculating such a term for every possible sequence of symbols, and adding up all the terms. This relationship can be simply expressed by an equation given in the notes.<sup>4</sup> If the logarithm is base 2, the sum of all the terms, one for each allowed configuration, is the amount of Shannon information characterizing that system, in bits.

The measure has two distinct interpretations. It all depends on your perspective. From the point of view of a communications engineer, the Shannon measure characterizes the capacity of a system to carry information. This way of thinking measures uncertainty—the greater the capacity of a system, the more you don't know about a particular message. For this reason, Shannon called his measure *entropy*, in keeping with the terminology used in thermodynamics.

From a different perspective, that of a particular object you are interested in, Shannon's measure can be seen as the amount of information carried (but not necessarily communicated) by that object. This is the interpretation mostly adopted in this book. Confusion arises because if you have a particular object (string of symbols) in hand, there is no uncertainty, the Shannon entropy (from the communications perspective)

is zero! However, if one considers a particular object in the context of all possible, but unrealized, configurations of the same system, then it makes sense to equate the information carried by a particular object to the entropy of the system to which the object belongs.<sup>5</sup> Alternatively, it is sometimes useful to simply define the Shannon “self-information” to simply be the negative logarithm of the probability of the object or the event.

A practical feature of the Shannon definition (in both meanings) is that if a language uses just two symbols (for example, 0 and 1) and the probabilities of all permitted symbol sequences are the same, then the information, in bits, equals the length of the message (or object). This means that a string of one thousand 0s and 1s (i.e., 10110100010... in any order) carries one thousand bits of information unless there is some rule biasing the probabilities of the 1s and 0s in the sequence. This special case illustrates how closely the Shannon information measure is related to message length.

If the state probabilities (those used in the equation given in note 4) are not all the same, then the information carried is less than the maximum amount of information it could carry. Probabilities, however, are context dependent. They depend on rules in force or what is known. For this reason, it often makes sense to take the position that you know nothing about the context and simply assume the probabilities of the various possibilities are the same. In this case the “information content” of an object equals the maximal entropy of the system (of which the object is a particular state) and is equal to the length of the message (in binary).<sup>6</sup> To make things simple, I will generally use length to characterize the amount of information a message may carry. When a message is written in binary, length is measured in bits. This is the measure used to characterize the capacity of a DVD or the size of a file in your computer.

Shannon’s formulation is a foundational concept in the modern communication industry, but it has some peculiar features. For example, when used to characterize objects, it does not discount redundancy (unless the redundancy is accounted for in the probabilities). Thus, the double message “*I went to the market I went to the market*” contains twice as much Shannon information as does the single phrase. In common usage this is silly. If you read something and remember it, reading it again does not double your knowledge. Another source of confusion comes about because the Shannon information measure has nothing to do with meaning. “*wtm naIt oeeKh t tre*” embodies exactly the same amount of

Shannon information as *I went to the market* because it is the same length and uses the same letters and number of spaces. Shannon's measure is based strictly on the number and probability of distinct messages that potentially can be encoded by a system, nothing more. A simple way to reconcile the measure with everyday use of the word *information* is to consider Shannon information to be the maximum amount of information that a message might contain. The amount that is actually meaningful or useful may be less. It may even be zero.

In physics a system of molecules has maximal entropy when the molecules are free to wander around in space exploring all available microstates. When structure is present, some of the molecules are not free to move just anywhere, they are constrained to particular locations. Thus a system with physical structure is characterized by less entropy (less freedom) than is a system with the same number and kinds of atoms having no structure (at the same temperature). This difference between the actual entropy and the abstract maximal entropy is sometimes called *negentropy*.

Structure also can be present in sequences of symbols. In this case structure refers to nonrandom distribution of symbols. Symbols don't move around in a sequence, but nonrandom patterns reflect unequal probabilities. If one can determine the actual probabilities, use them to calculate the Shannon entropy, and then subtract this value from the maximal entropy (probabilities all the same), the resulting value is the information science equivalent to physical negentropy. It measures the amount of structure present.<sup>7</sup>

Any system with multiple parts can be characterized in terms of entropy and information. Importantly, there are other information-like ways of characterizing systems. For example, when considering systems with obvious internal structure, it is quickly apparent that some are more complicated than others. This aspect of structure is loosely referred to as complexity and is not fully captured by Shannon information, entropy, or negentropy concepts. We will discuss more about this later.

## What is an evolutionary system?

All evolutionary systems rely on stored information, and all modify, add to, or delete from this body of information by following a well-defined

information processing strategy. At the core of every evolutionary system is a probabilistic computation that has the remarkable property of extracting purposeful information from randomly occurring events. When this computation is employed to assemble instructions for making something useful, a positive feedback loop can be established in which any change in the instructions that causes an improvement in the structures or actions specified by the instructions serves as the basis for future improvements to the instructions and their outcomes. By this means, instructions involved in this virtuous cycle get better and better over time at whatever they specify. Significantly, when the instructions specify the solution to a problem, this strategy is the most powerful method known for finding solutions to problems for which one initially has no idea about what a good solution might be.

When objects and activities involving objects are framed in informational terms, it becomes clear that randomness is the ultimate source of anything truly new. This may seem startling to those not familiar with the foundations of computer science, but the logic behind it is sound. Deterministic rules precisely determine outcomes; therefore, in the absence of some form of random input, outcomes are always predictable. Deterministic computers are so useful because they output the same answers every time they are given the same input. This is the opposite of creativity. The notion of creativity implies unexpected outputs, but such outputs are only possible in a deterministic process if unpredictable inputs are allowed. Unpredictability is the essence of randomness. Provision for random input is incorporated into the design of all creative activities, and evolutionary processes of the sort discussed in this book can almost always be seen and analyzed as probabilistic computations that draw directly from sources of randomness.

The general features shared by all evolutionary processes are familiar to biologists and to many engineers, and always include the following five elements<sup>8</sup>:

1. Entities. There are always multiple units of some sort. These have various names including but not limited to: organisms, agents, genes, concepts, and corporations.
2. Heritable features. Aspects of each entity are specified by information encoded somehow as a part of the entity (in biology this information is encoded in DNA).



3. Entities reproduce or are copied. Because of this, the encoded information associated with each entity is a copy of similar information present in parental or precursor entities.
4. Change mechanism. Information copying, reproduction, or maintenance must provide an opportunity for modest change. In many systems the change mechanism is simple error during copying.
5. Feature-based selection. Reproductive (or copying) success must be determined, in part, by features specified by each entity's encoded information.

***Whenever these five elements are found in the same system, the encoded information and corresponding entity features present in entity populations will change over time; and inherited entity properties will inevitably become more and more adapted to whatever criteria determine reproductive (copying) success.***

Systems embodying this logical structure accumulate encoded information relevant to satisfying the selection criteria. They do this by accessing the raw potential of random change, and sometimes nonrandom change, while avoiding the overwhelming odds that work against probabilistically generating something useful or interesting all at once. This process of information accumulation and modification is a probabilistic computation. The reason it works is simple. ***Very small changes are usually not at all improbable.*** The smallest possible changes are of the yes/no, up/down, either/or type. Such choices are typically characterized by 50/50 odds of one or the other possibility happening, or of one being better than the other. In contrast, undirected large changes are wildly improbable. By sampling small, not very improbable changes, and accumulating the better ones, outcomes that are otherwise hopelessly improbable can be achieved easily. This is what Richard Dawkins, the most famous living writer about evolution, meant by the title of his book *Climbing Mount Improbable*.<sup>9</sup>

Dramatic examples of change can and do occur when repeated error-prone copying of instructions is coupled with a mechanism for choosing which of several outcomes are to be copied again. When such a sequence of selective copying becomes established, complex highly improbable purposeful instructions can be assembled over time even though no direction is provided other than the conditions that establish which instructions are copied. Complex instructions, in turn,

make possible the formation of complex and otherwise highly improbable objects or behaviors. Later we will examine examples of structures, some fairly complicated, that form spontaneously as a direct result of the laws of chemistry and physics acting on naturally occurring matter. Such objects do not require instructions. Formation of non-instruction-based structures is limited to things that can form spontaneously with reasonable probability. Use of instructions, in contrast, offers unlimited possibility for the creation of useful, wonderful, clever, beautiful, and otherwise hopelessly improbable things. We will examine some examples of these in chapter 4. It is no exaggeration to say that ***anything permitted by chemistry and physics is attainable through the use of appropriate instructions***—quite a remarkable statement. The challenge, of course, is to get hold of the right instructions. In the absence of instructions, possibilities are much more limited.

Because instruction use is so powerful and so pervasive, it is useful to distinguish the complexity that results from the use of instructions (Type II objects) from complexity that does not require instructions (Type I objects) and to attribute a higher form of complexity to Type II things. I assert repeatedly in this book that all but the most trivial instructions are created in an evolutionary computation (we know of no other computational strategy capable of doing the job). A test of this assertion is that whenever exceptional complexity is encountered, one should be able to identify the explicit form of the encoded instructions and the physical basis of the computation that has created them. For living organisms, this information is encoded in DNA molecules. Cells and organisms carry out the computations by living and reproducing. Actually, living organisms depend on two very different computations. One is the creation and maintenance of the organisms themselves. These processes are based, in part, on information encoded in DNA.<sup>10</sup> Not everyone would consider these chemical and developmental processes to be computations at all. A quite different computation is the slow modification of DNA-based information measured in generations that occurs over time. This is the evolutionary computation this book is about. A branch of biology known as population genetics describes how life-modifying evolutionary computations proceed over generations.

Deterministic electronic computers are capable of performing evolutionary computations, but not the same ones performed by life.

With electronic computers, the evolving body of information is computer code. Your brain can also be considered to be a kind of computer where information is encoded in patterns of nerve cell communication. In chapter 10, we will explore the possibility that the defining activities of learning and thought are evolutionary computations.

Viewing evolution from the perspective of information accumulation and refinement permits deeper understanding. It also allows some otherwise difficult issues to be approached more easily. For example, the notion of evolutionary progress is controversial in biology, and then too, no one disputes that the ultimate origin of life is an unresolved scientific puzzle. The computational perspective helps to clarify both of these issues. In the case of evolutionary progress, if one equates advancement to increasing complexity, then information theory allows the identification of specific conditions under which ever-increasing complexity is an expected outcome. Later we will discuss one particular aspect of complexity, depth, and show why it is expected to increase under certain conditions. Toward the end of the book we will consider how the origins of all evolutionary systems including life are constrained by information theory. More broadly, because evolutionary computation has such enormous potential for generating ever-greater complexity, it is tempting to suspect the evolutionary engine to be at work whenever great complexity is observed.

## What constitutes scientific understanding?

My goal in writing this book is to provide the reader with a deeper understanding of our world, including ourselves. I've worked hard to make my explanations believable and satisfying. Conclusions follow logically from observation and theory, but are not to be believed unless they can be tested by experiment. Not coincidentally, this is precisely how science works. Each reader can make his or her own evaluation of whether the arguments and tests are convincing.

At a very basic level, science is possible because our universe is characterized by predictable patterns. The central activity of science is discovering these patterns. Not everything is orderly, far from it, but order is definitely out there. Newton's universal law of gravity (objects with mass attract one another with a force inversely proportional to

the square of the distance between them) describes such a regularity.<sup>11</sup> Before Newton, no one recognized the exact nature of this pattern even though everyone knew that things fell down, not up, or sideways.

However, scientific understanding involves more than a list of observed regularities. To claim that we understand some structure or activity, we must not only identify relevant laws but also be able to show how they play out over time in specific situations. For example, engineers can use the law of gravity, the conservation of energy, and the concept of momentum to predict the trajectories of artillery shells or the stresses experienced during a roller coaster ride. When such predictions are confirmed, we gain confidence in the principles used to make the predictions. Science is held in such high esteem precisely because so many science-based predictions are confirmed every day.

Mechanisms illuminate principles and confirm understanding. When scientific understanding leads to the creation of a device that produces a predicted structure or action, understanding is confirmed and enriched. Consider the theory of sound waves. Theory holds that sound consists of waves of alternating compaction and rarefaction of molecules making up the transmission medium, usually air. Loudspeakers and microphones are built to exploit this principle. A loudspeaker contains a flexible surface that is driven back and forth by some device, typically an electromagnet. Rapid movement of the surface alternately compresses and decompresses the air in front of it, setting pressure waves in motion. But how do we know that the waves actually exist? A microphone is a device built specifically to detect pressure waves. Incoming pressure waves physically deform a flexible surface. These motions are transduced into electrical signals that can be amplified to drive another loudspeaker, displayed on an oscilloscope, or recorded for future use. The design and actions of loud speakers and microphones make no sense unless the theory of sound waves is essentially correct. The devices we build both illustrate and confirm scientific theory.

Science offers two compatible but largely independent bodies of understanding to explain physical objects. These are physical theory and biological evolution. Physical theory is devoted mostly to understanding nonliving phenomena but also permits insight into how organisms function. Biological evolution explains how living organisms came to be and why they are adapted to their circumstances.

Traditional evolutionary theory offers little for the understanding of galaxies or atoms. Over the past fifty years or so, however, the scope of evolutionary explanations has gradually widened to include things other than living organisms including human made artifacts and certain classes of computer programs.<sup>12</sup> A few physicists have even proposed ideas for how the cosmos may have evolved in ways reminiscent of biological evolution.

Three hundred years ago Isaac Newton explained the complexities of our solar system in a new way by invoking a few simple principles and analyzing the consequences with a new kind of mathematics called *calculus*. Since then, enormous progress in science has led to the discovery of new principles and from them new explanations for many aspects of our physical world. Today biochemists, molecular biologists, and physiologists study how the laws of chemistry and physics play out in living organisms.

One-hundred and fifty years ago Charles Darwin explained the complexities of life in a new way. Much is made in the popular press about evolution being a theory while physics is based on laws. But in science, a theory is simply an explanation of a phenomenon. Depending on how intensely they are tested, theories can be tentative or securely established. The term *law* is less well defined in science, but laws can also be tentative or well established. Laws tend to describe relationships or principles. Usually a law can be expressed as a short statement or a simple equation. Newton's law of gravitation expresses in algebraic terms a relationship between two masses. The first law of thermodynamics (energy can be neither created nor destroyed) expresses a principle.

Evolution is neither a relationship nor a simple principle. Rather, evolution is a process that creates a progression of outcomes each based on a previous outcome. Because of this, ***evolution is rarely expressed as a law even though it is a powerful and widely accepted scientific explanation.*** Like all physical processes, evolutionary theory is based on measurement, mathematical truths, and physical laws. Evolution is hardly the only theory that explains outcomes of processes. Star formation, continental drift, and an engineer's explanation of how the internal combustion engine works are also process theories. Successful explanatory theories rest solidly on physical laws and careful measurements, but what they capture is often not expressible in simple equation form.

Contrary to claims made by some antievolutionists, science is not limited to subjects that can be tested directly in the laboratory. It is common for some phenomena to be considered well understood even though they cannot be reproduced by human effort. Consider, for example, mountain ranges, stars, and fossil dinosaur eggs. Explanations for physical phenomena are accepted when they flow logically from previously accepted theories, principles, and laws and when plausible mechanisms can be imagined, simulated, or demonstrated. Evolutionary explanations of organisms and features of organisms that originated long ago are accepted by scientists in this way—as are scientific explanations of many other things that are either too big or too slow forming to be reproduced in the laboratory.

The theory of evolution has been criticized for not offering explicit proof that major evolutionary inventions such as focusable eyes, bacterial flagella, or the complexities of cellular metabolism originated through small sequential changes over eons of time.<sup>13</sup> Many of the arguments were subsequently shown to be based on misunderstandings or on an incomplete reading of the scientific literature. In the last two decades, rapid advances have been made in understanding how regulatory genes shape animal organs and appendages. The emerging principles are making it clearer year by year that through small understandable steps the process of evolution is able to gradually create totally new organs and organisms. A new field of study, nicknamed “EvoDevo” (evolutionary developmental biology), has emerged that reveals detailed molecular mechanisms showing how existing biological structures were created by incremental changes over time through random mutation and natural selection. Key to this process is the cooption of existing structures for new purposes. Nonmicroscopic examples have long been known: fish fins over time becoming legs for walking, forelegs becoming wings (in bats and birds), jaw bones becoming inner ear bones, and leaves becoming protective spines (in desert plants). More recently, analysis of DNA sequences has revealed many times over that macromolecules playing key roles in complex organ formation today once contributed in modified form to some other seemingly unrelated structure in the distant past.<sup>14</sup>

Essentially all biological scientists living today accept the theory of evolution as the correct general explanation for living complexity even though no one yet has artificially created a living cell from nonliving

matter or a complex structure such as an eye or an ear by an evolutionary method. Tens of thousands of experiments have confirmed the power of selection acting over time to change populations of organisms ranging from viruses to milk cows. The adaptations of organisms to their environments and the relatedness of DNA in one organism to DNA in others are inexplicable without the theory of evolution. Successes of modern plant and animal breeding feed much of the world and testify to evolution's essential correctness. It does not take much imagination to project what might be accomplished by a plant or animal breeder if an artificial selection program could be carried out for a million years.

Scientific understanding does more than provide explanations of observed phenomena; it also enables otherwise unimaginable technological advances. Many of these have transformed human life and the planet on which we live. If you doubt this, consider that 400 years ago the total number of human beings was less than one-twentieth that of the modern era. Human populations at that time were far more limited by disease and famine; life expectancy for most was less than thirty years. The typical person farmed the land and was guided at night by moonlight or oil lamp. He or she traveled by foot, horse, or wind-powered ship. Communication between distant individuals was limited to those few who could employ messengers. The difference between then and now is almost entirely technological. True, there have been advances in institutions and social structures that have improved the lives of many, but most of these changes have been enabled by new technologies; and no one can dispute that over the last 400 years advances in science-based technologies have utterly changed our world.

Humankind has been creating technologies since long before recorded history, but until recently progress was very slow. What changed the pace of innovation? The answer is simple: once relevant natural laws and principles were understood, new mechanisms could be invented that made use of the principles. Spaceships to Mars would not be possible without an understanding of gravity and of the chemical reactions that propel rockets. Exactly constructed devices such as space vehicles that do what we intend them to do are not found in nature. Science explains existing phenomena, and technology uses this understanding to create new mechanisms. Because of this, technological achievements confirm by their very success the correctness of the

scientific understanding on which they are based. If Maxwell's theory of electricity and magnetism were not correct, electric generators as currently designed would not produce electricity and electric wall switches would not turn on lights. Every time we flick a light switch we are confirming scientific understanding of electricity. When scientific understanding is wrong, engineers cannot use it to construct devices that work. This is why astrology, alchemy, witchcraft, and religious dogmas are of no use for practical matters such as building bridges or curing diseases, and why they have no credibility within science. To reiterate, technology is possible without science, just more limited; and when science-based technology works, it confirms the science on which it is based.

Given this discussion, is there a simple answer to the question: What is scientific understanding? Yes, *scientific understanding constitutes the current consensus among all those who have thoroughly studied a topic*. Such understanding is constantly being probed and tested by new experiments designed to explore whether logical consequences of that understanding lead to expected outcomes in new situations. When they don't, something is wrong, and either the logic of the experiment needs to be rethought or the basic understanding needs to be modified. Turning this simple principle around, if anyone wants to challenge current scientific understanding, all they need to do is publish a reproducible experimental outcome or a logical analysis that is incompatible with current understanding. The scientific community will then scramble to further reproduce the experiment or confirm the logic and, if the new experiments hold up, the old understanding will be modified. Through this method, scientific understanding is constantly changing—or could we say evolving?

### How does computer science fit in?

Since Galileo's time the accepted language for characterizing natural regularities has been mathematics. Our understanding of physical phenomena is written mainly in the language of mathematics, and mathematics is widely used to predict detailed outcomes. That biology and social science do not seem to follow this dictum is usually attributed to their extreme complexity. Many authors have predicted it will only



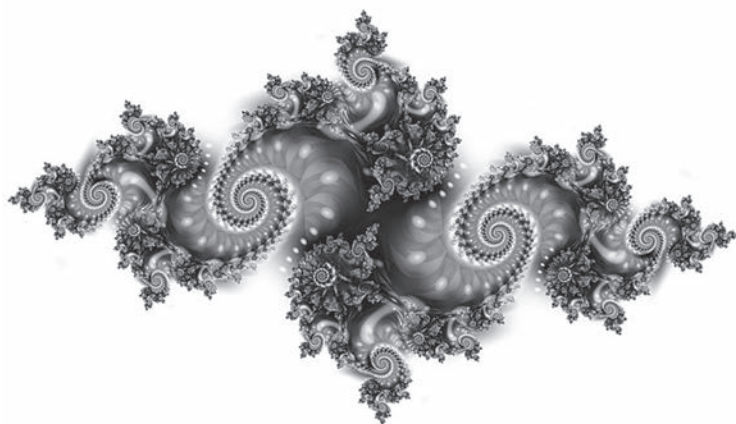
be a matter of time before the proper mathematics is found and used to support new fields of biological or social engineering. My view as explained in this book is that the proper mathematics has already been found: it is computer science.

Computer science and its associated body of theory provide powerful insights into the usefulness of traditional mathematics for understanding complex processes. Computer input can be divided into program and data, where the program constitutes rules specifying how the information inherent in data is to be manipulated. Results of that activity are called outputs.

The physical world can be formalized in a like manner. If the laws of physics are the universe's program, then physical structures are input and output. A snowflake is the consequence (output) of the laws of chemistry and physics acting on a below-freezing atmosphere containing an excess of water vapor (input). We can think of physical processes as computations in the general sense that input configurations are acted upon by the laws of nature to yield output configurations.

Computational theory reveals aspects of relationships between input and output that are fundamental for understanding all kinds of complex things. One very important principle is that remarkably complicated things can result when simple rules act on simple input. This is especially true when each output serves as input to another round of computation. With appropriate rules, the longer such a repetitive process operates the more complex the output may become. Good examples are provided by computations that produce intricate visual patterns by repeatedly applying the same simple operations over and over using the output of each round as the input to the next. Figure 1.1 shows a picture resulting from such a process.<sup>15</sup> Much in the physical universe can also be interpreted as resulting from repeated application of simple rules. Snowflakes, galaxies, and volcanoes are complex outcomes resulting from the actions of simple physical laws applied over and over.

A less obvious, but equally important principle of computer theory is that some complex things *cannot* be created by employing simple rules no matter how long a process is carried out. Such objects are achievable only if complicated rules, that is, long programs, are employed. Because the laws of chemistry and physics, and indeed, most things taught in traditional mathematics, deal with relatively simple relationships, computer theory tells us we should expect structures to occur whose



*Figure 1.1.* Computer drawing generated by the repetition of a few simple rules.

formation cannot be expressed in simple mathematical terms. Modern digital technology offers numerous examples. Consider the computer program used to create this document. Microsoft Word offered various logistical supports including saving, formatting, tracking, grammar checking, and spell checking. The basic program file instructing my computer how to do these things required 8.4 megabytes (MB) of hard drive space. Thus, the “rules” used to create the formatted document from my typed words are longer than this book (roughly 1 MB of text). Simpler word processors have shorter programs, but none can be reduced to a handful of formulae. No branch of mathematics offers tools that calculate the formatting of a book. The simplest way to format is to follow a long list of rules, like a chef preparing a soufflé. The recipe discussed earlier provides a tangible example. It is just not possible to write a simple formula for a soufflé.

When relationships between numerical input and numerical output can be expressed in mathematical terms, the output can often be calculated. The power of mathematics is that it provides shortcuts. When simple relationships exist, it is often the case that the physical process of interest does not need to be carried out to determine a particular outcome. An example is the calculation of future positions of planets. When the relationships are not simple (the program is long), there are typically no shortcuts, and a process must actually be carried out if the

outcome is to be learned. Computer science does allow for the simulation of natural processes, but if the conditions are complex and the process long, accurate simulations require computations that are fully as complicated as the phenomena they are simulating. Some examples in this category are the weather, the economy, and the performance of airplane wings at high speed.

When one considers *all possible relationships* between inputs and outputs, those with complicated relationships far outnumber those with simple relationships. No one knows if this mathematical truth is reflected by the structure of the physical universe, but it does suggest that we might expect to find phenomena that are not understandable in simple mathematical terms. Living organisms, ecosystems, and most societal activities probably fall into this category.

In most discussions of rules, the focus is on deterministic rules where the same initial conditions always result in the same output. However, most physical processes and some computer programs obey rules that allow random events. If random actions are possible, then specific outcomes *cannot* be predicted even if the rules are simple. When such a process is carried out repeatedly, the result is different each time. Fortunately, in many of these situations statistically averaged predictions are possible. Related to this is what we might call the general prediction. Snowflakes provide a good example. It is not possible to predict details of the next snowflake, but snowflake properties are quite predictable in general. In some ways, biological evolution is similar. The rules governing the process of evolution admit certain kinds of random change, and broad trends can sometimes be predicted or understood even though specific outcomes in the past or future cannot.

### How is purposeful complexity explained by science?

Earlier we saw that one can divide all objects into two categories: Type I based on simple rules, and Type II requiring additional complicated rules, or instructions. By making this division, the exceptional complexity characterizing the planet Earth is seen to consist entirely of Type II objects, their behaviors, and their consequences. The division focuses our attention on instructions and also requires that we explain where the information in instructions comes from. Type II objects have

another characteristic: they can always be seen in terms of some “purpose” or “fit.” Instructions always have a purpose. Every instruction is directed toward the creation of a specific outcome, and the outcome can always be described as being for something or to fit some local circumstance. Daniel Dennett calls this way of talking and thinking about inanimate objects in human motivational terms the “intentional stance.”<sup>16</sup> He observes that attributing intent to nonsentient, nonthinking objects often leads to correct inferences about how they will behave in the future and to understanding why they are the way they are. Why do birds have wings?—to fly. Why do they fly?—to make a living and escape from predators. Why are overcoats well insulated?—to keep your body warm. Why do fish have muscular tails?—so they can swim fast in water. Each of these examples involves some fit to the environment. The number of potential examples is endless. Why are so many things this way? The answer comes from understanding how such things come to be. In other words, the origin of purpose is to be found in the processes that create Type II objects.

Despite the impressive potential of instructions for producing complex structures, most structures in our universe form without them, and some of these are vastly complicated. One need only consider a cluster of galaxies. In chapter 3 we consider some others, but none can reasonably be described as having a purpose. Another thing that sets instruction-based structures apart is the nature of their formation. They require specification that is permitted by, but not provided by, the laws of nature. Instructions don’t violate the laws of chemistry and physics but rather take advantage of them to make specific structures or processes happen that otherwise would not come about.

Appearance of purpose has long been an issue in both religion and philosophy. It was the basis for the Rev. William Paley’s famous 1802 argument for the existence of God, the creator. How, he argued, could there be purposeful objects (including people) without a creator to impart purpose? The counterargument that evolution can provide the appearance of purpose is presented powerfully and eloquently by Richard Dawkins in *The Blind Watchmaker*.<sup>17</sup> He argues that apparent purpose is established by selection. Selection is an essential aspect of the evolutionary mechanism. It is responsible for choosing which variations in the current generation will populate the next. Over time individuals comprising a population will exhibit more and more features that reflect

past selections. A fish appears designed to live in water because its ancestors thrived better in water than those that left no progeny.

Gary Cziko<sup>18</sup> in *Without Miracles* also addresses the question of how complex objects come to be. He explains a broad sweep of complex phenomena ranging from bacterial evolution to human thought, essentially everything that is characterized by a good “fit.” There are two kinds of fit and one is trivial, the kind produced when a china dish is broken. The resulting pieces fit together because they were derived by simple means from an original whole. The other kind, the one that provokes Cziko’s interest, is purposeful fit. This fit is achieved in the absence of an original whole. In such cases, something new is created to achieve a fit. Examples include the fit of antibodies to viruses, the fit of organisms in their ecosystems, the fit of technological inventions to the needs of their users, and the fit of ideas to the desires and needs of an individual. He argues that repeated selection is the only process capable of explaining why complex things (Type II objects in our terminology) always make sense in terms of some local context.

By definition, instructions and Type II outcomes are always paired, and instruction-outcome pairs are always motivated by something. Objects that result from following instructions fulfill the purpose of the instructions, and instructions are created in response to some need, desire, or potential. An example discussed in chapter 4 is a screwdriver. Screwdrivers are useful because they fit both the head of a screw and the human hand. What sets screwdrivers apart from rocks and other Type I objects, even potentially useful ones, are construction details too improbable to happen in the absence of instructions. In this case, the instructions are created because people want to turn screws. Fish fit their watery environment so they can live and reproduce. The motivation in that case is continuation.

Instructions do not just happen; they require explanation. In scientific study, to state that a person (or God) thought them up is not acceptable. Instructions embody specific information, and we must identify the nature and sources of this information if we are to achieve a deep scientific understanding of our world. The only known procedure capable of producing instructions when the method is not known in advance is evolutionary computation. When making this claim, I am not speaking specifically of the evolution of life, but rather I am taking a general view in which evolution encompasses a class of activities that

accumulate useful information through a particular computational strategy. Underlying all evolutionary processes, whether involving life or not, is a computational cycle that extracts purposeful nonrandom information from small and often random changes. The computation works by making minor changes to an existing already useful body of information and by retaining those changes that provide added usefulness as judged by consistent criteria.

I stated earlier that the complexity of a Type II object stems directly from the complexity of its instructions. A soufflé is more complex than scrambled eggs precisely because there are more steps required for its creation. These steps are listed in the recipe. In general, the more complicated the set of instructions, the more complex the outcome may be. Instructions specify a process and the resulting output of the process usually exhibits more complexity than would be possible if the process were simpler.

Multiple-step processes are by their nature improbable, and the longer the set of instructions needed to specify a process, the more improbable it is. Closely tied to this is the observation that with appropriate instructions and a means of implementing them, there is no limit to how complicated (and how improbable) the resulting structures can be. Consider a modern computer. The creation of this machine, a physical structure, requires that ores be mined, semiconductors and metals refined, and plastics with various properties created. These materials are used to fabricate intricately designed “chips” and to manufacture numerous other structures including wires, transformers, and fans. The components must then be assembled in a very precise manner. Instructions specify everything: the exact composition and design of each part, the location of each part, the order in which the parts are added, and the connections between the parts. Only the laws of chemistry and physics are not specified. Without instructions, electronic digital computers are quite impossible. When followed, instructions lead naturally to a predetermined final structure. This “instruction” principle underlies all modern technology and manufacturing and also all significant artifacts of human culture including music, architecture, science, even the institutions of religion. Living organisms depend on the same principle.

In evolutionary systems, changes occur over time measured in generations, and purpose is established slowly, tiny step by tiny step, through the process of cumulative selection. To illustrate, fish live in water, and

their DNA encodes information used by the molecular machinery of growth and development to form a hydrodynamic body shape with a muscular tail. This information, stored in DNA structure, has the purpose of allowing fish to live and move efficiently in water. However, it was not always so in the history of fish. Once upon a time fish ancestors were very small and did not swim by muscular action; rather, propulsion was by means of cilia, protrusions from cells that act like tiny oars. Those creatures had DNA, but that DNA encoded little or no information concerning hydrodynamic shape or tails or muscles. Such information is encoded in fish DNA today because past changes in DNA were favored when they resulted in a better fit of fish to their environment. Proto-fish ate small things, and there must have been an evolutionary advantage for being bigger. As mutations occurred that made some offspring larger, the larger individuals thrived and, in turn, had more offspring. But largeness presented other problems. Large bodies presented greater drag, and cilia became less effective as a way of moving about. Over time wiggling motion supplemented ciliary action and this permitted still larger bodies. Swimming became more and more efficient under the relentless pressure of natural selection. Eventually, ciliary motion was abandoned altogether in the fish lineage.

Efficiency provides one measure of fit that often translates into higher reproductive success. When initially random changes to DNA happened to result in greater efficiency, those changes were perpetuated in DNA structure. Through the tedious repetitive process of trial and selection, purposeful information about hydrodynamic shape, tails, and musculature slowly accumulated and became perfected in fish DNA. The process began with DNA that encoded no particular information about efficient swimming and has now reached a state where lots of pertinent information is present.

When viewed from a computational perspective, the scientific explanation of purpose is found in the way instructions are created. The flip side of this understanding is that objects not requiring instructions have no purpose. They may be useful for something, but they are not made *for* something. Instructions, by their very definition and because of the way they are formed, are purposeful, and so are the things they specify.

Before we more fully explore the origin of instructions, the evolutionary computation I call *the engine of complexity*, we need to further explore the nature of computation and relationships that exist between computation, physical process, and the formation of structure.

---

## Computation

### What is a computation?

Listen to music or surf the Internet on your iPad. Start your car in the morning without touching the accelerator pedal and your engine “decides” how much gasoline and air flow into the cylinders; as the engine warms, the ratio of gas and air changes. Step on the accelerator and the ratio changes again. How are such things possible? Electronic computers function in each device. It is pretty hard these days to escape the influence of computers. They calculate our bills, produce photographs without film, keep track of your likes and dislikes, and predict the weather. Computers are machines that manipulate information, and computer science is the formal discipline that studies limitations and opportunities afforded by this type of activity. In this chapter we will explore the concept of computation. I hope to convince you that the notion of computation is more general than simply what happens in electronic computers and that it is not possible to cleanly separate many kinds of physical activity from computation.

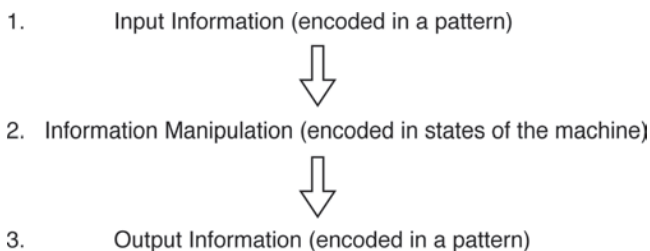


Basic to the discipline of computer science is the idea that information can be encoded in patterns of symbols, usually linear sequences that convey meaning to someone or something. Chapter 1 introduced one specific quantitative measure of information: Shannon information. This measure is very useful for some purposes, such as determining how much physical space you need to store some information you value or how much bandwidth you need to transmit it to a distant location. Later in this chapter I will introduce another measure, algorithmic information, which is particularly useful in computer theory.

A convenient way to understand the notion of computation is to see every computation as a process in which a pattern (usually, but not always, a sequence of symbols) interfaces with a device in such a way that a series of changes occurs within the device, culminating in output. The output may be useful in ways that the input is not. This general idea framed in terms of information is diagramed in figure 2.1.

This raises several questions: What does it mean to manipulate information? Is information preserved during a computation? Where does new information come from? What are the minimum requirements for a computer to be able to “manipulate information”? To answer such questions, we need to develop some background.

Let us start with the idea of a “system” in physics. A system is anything physical that has a boundary. Empty space is not a system; neither are certain abstract ideas such as a mathematical point or a geometrical plane. The universe may or may not be a system depending on whether it has a boundary. All things smaller than a finite universe and larger than a subatomic particle are systems and are also parts of larger systems. Multiple parts raise the opportunity for different arrangements



*Figure 2.1.* The three parts of a computation.

of those parts. We call any one arrangement a “state” of the system. In chapter 1 we saw that any time states can be distinguished from one another, they potentially carry information. Whether or not that information is good for anything is another matter. Relevance depends on whether there exists a second system that is changed in specific ways by interacting with specific states of the first system. Think of driving a car and imagine yourself as one system and the car as a second system. These two systems interact. You, seated in the driver’s seat with your foot on the gas pedal, are in a state of the “you” system. Press your foot down and you are in another state; this new state causes the car system to transition to a new state in which it is going faster. Certain of your personal states have meaning to the car system; others, such as closing one eye or waving your left hand, do not. In the world of physics, meaning emerges from interactions between system states. If there are no interactions, there is no meaning. For meaning to be present, particular states of one system must have particular effects on another system.

In the world of information science, meaning is often tied to the notion of representation. A principle that underlies the whole concept of computation is that *one state can be represented by another state*. The states need not be in the same system. To give two examples: the words you type on a keyboard can be represented by voltages and current flows inside an electronic computer; music that is performed by a human artist can be represented by a pattern of silver and black dots on a DVD. When a particular representation affects you, it has meaning to you.

Computer programs consist of commands that cause specific sequences of states to occur inside a computer resulting in output. In computer science, a sequence of commands that accomplishes something is called an algorithm. In other venues they may be called recipes or instructions. Instructions and algorithms represent methods for accomplishing tasks.

Most computer algorithms accept data allowing the output to be tailored to specific situations. Both data and algorithms embody information. Data can be thought of as a representation of some aspect of a world outside the computer. In this way of thinking, a computation consists of a representation of a method (an algorithm) acting on a representation of some aspect of the world (the data) resulting in a

sequence of internal representations (states of the machine) that culminate in a final representation (the output).<sup>1</sup> Each of the various representations is a physical state of a system, and all are characterizable in terms of information. This may seem pretty abstract, but remarkably, it can be implemented in physical devices and ultimately explains why information is so important to us in so many venues. Computers are devices that can take on a multitude of internal physical states. Within them internal states interact with input states creating new internal states. The final state is the output.

The modern concept of computation has developed over many decades. Before there were computers, people computed answers to math problems, and mathematicians and philosophers alike were curious about how they were able to do this. In the late 1930s Alan Turing, a young British mathematician, established the mathematical foundation of modern computer science. A few years later he led a successful British effort that broke the German military codes. In a little-known footnote to World War II history, Turing's critical contributions to the war effort were so secret that he had difficulty getting a job after the war. It was only after his death by suicide in 1954 that he became widely recognized as a war hero. Turing's greatest contributions to science began with a careful analysis of what a person does when he or she solves an arithmetic problem. Consider the problem of dividing 58432 by 83. For most people, solving this problem without a calculator involves performing a prescribed sequence of steps, each one of which is quite simple. Most of us were taught how to do this in elementary school. Listed in box 2.1 are twenty mechanical steps that if followed will produce the correct answer to this division problem. The tediousness of these steps is well remembered by those who, like me, hated long division in elementary school. If you don't feel like reading each line of the box, that's OK.

Anyone who follows the sequence of steps will arrive at the correct answer. It is not necessary for them to "understand" what they are doing or have any talent for guessing well in steps 3, 11, and 14. They do have to be able to determine whether one number is larger or smaller than another, and to multiply and subtract. If they don't know how to multiply, multiplication can be reduced to repetitive addition (increasing the number of steps). Turing argued that any computational problem

### Box 2.1

Twenty simple steps that if followed solve the problem  $58432 \div 83$ .

1. Write the number 58432 on a piece of paper.
2. Determine whether 83 is smaller than 5, 58, or 584.
3. Since 83 is less than 584, guess the largest number less than 10 that when multiplied by 83 yields an answer less than 584.
4. Write this number above the 4 in 58432.
5. Multiply the number just written by 83.
6. Write this number below 584 (in 58432) and subtract.
7. If the answer to step 6 is negative, erase the number written in step 4, reduce it by one, and proceed again with step 4.
8. If the answer to step 6 is positive and equal to or larger than 83, erase the number written in step 4, increase it by one, and start again at step 4.
9. If the answer to step 6 is positive and less than 83, append a 3 (from 5843) to the right of it.
10. Compare this number to 83.
11. If it is greater than 83, guess the largest number less than 10 that when multiplied by 83 yields an answer less than the number.
12. Write this number above the 3 in 58432.
13. If the number in step 10 is less than 83, write a 0 above the 3 in 58432 and append 2 (from 58432) to the right of the number created in step 9 (since the subtraction in step 9 results in a number less than 83, we will omit the steps that would follow a different outcome).
14. If the number created in step 13 is 83 or larger, guess the largest number less than 10 that when multiplied by 83 yields a number less than or equal to it.
15. Write this number above the 2 in 58432.
16. Multiply the number written in step 15 by 83, and write it below the number created in step 13.
17. Subtract.
18. If the number from step 17 is negative, return to step 14 and reduce the number by one.
19. If the number from step 17 is positive and equal to or larger than 83, return to step 14 and increase the number by one.
20. If the number from step 17 is zero, you are finished and the correct answer (704) is written above 58432.

that can be solved, no matter how complex, can be solved by following a sequence of extremely simple steps—steps even simpler than those in box 2.1. Since steps that are simple enough require no insight or thought, a (nonthinking) machine could be designed that would carry them out. He then showed how to design a machine that could do it. His original concept is now called a Turing machine in his honor and is taught to all beginning computer science students. Modern computers have different designs but always reflect Turing's concept. Turing machines are capable, in principle, of solving any computable math problem by proceeding through an appropriate sequence of mindlessly simple elementary steps.

The first practical electronic computers were built during World War II and used to calculate trajectories of artillery shells by solving repetitive numerical problems and to break secret military codes by systematically, and mindlessly, testing millions of possibilities. Regardless of their specific designs or the details of the computations they perform, all electronic computers carry out long sequences of very simple steps. They are useful because they perform these steps accurately and fast.

The twenty steps listed in box 2.1 constitute an *algorithm*. A remarkable feature of algorithms is that their logical structure transcends the physical system they are encoded in. This means that the outcome of an algorithm is independent of the device used to perform it. To quote Daniel Dennett, the solution to a long-division problem is the same if solved “on paper, on parchment, on a computer . . . or in skywriting.”<sup>2</sup> This is because an algorithm can be represented in different physical media, even in different languages. The transcendence comes about because symbol sequences can be copied, rearranged, and translated into other symbol patterns without losing the meaning or logical structure that characterizes the message. An important requirement for an algorithm is that it must have an outcome. A series of commands that does not yield output is not an algorithm and has no meaning.

Electronic digital computers accept programs (algorithms) written in languages that consist of linear sequences of symbols. The order of input symbols causes the machines to do specific things. For example, in many computer languages the command “ $x = 3$ ” will cause a computer to store the number 3 in memory with the label  $x$ , and a subsequent command “print  $x$ ” will cause the computer to send the contents of  $x$  (i.e., 3) to the printer or screen. The order of the symbols

is everything; neither “`rpint x`” nor “`x print`” will result in the number 3 appearing on the screen or printer paper.

The concept of computation is not limited to information encoded in linear symbol sequences. Two- and three-dimensional patterns can also be characterized in informational terms. People, however, including computer scientists, deal much better with linear systems; so nearly all human-made computers are based on the manipulation of linear sequences of symbols.

### What does it mean to manipulate information?

During the process of a computation, a computer creates a sequence of internal states, each determined by previous machine states and possibly by new input. The chain of states starts with input data, the algorithm (rules), and the initial state of the machine. Every aspect, input, rules, internal states, and output encode information. Because information ultimately resides in configurations (i.e., states), a computer can be seen as a device in which one state (the input) interacts with another state (the current machine configuration) to produce a final state (the output). Each state represents something, and it is all very prescribed. Accordingly, the simplest of all computations involves only two states, initial and final.

To do what they do most computers proceed through sequences of machine states determined by the algorithm and the input data. A word processor illustrates the concept. I type in letters and spaces one at a time (input), and I see them displayed on the screen in lines and pages bounded by margins (output). Interaction of the existing state of the machine with each keystroke causes new memory and processor states to form that in turn lead to other states that result in a particular state of a screen. Change the input and you get different internal states resulting in different output, but always the process is completely determined by the input, the program, and the design of the machine.

The general notion of a computation encompasses a wide range of activities, and it is convenient to distinguish three broad categories: reversible, deterministic, and probabilistic. Each has distinctive characteristics. Your personal computer is deterministic and not reversible. Because it is deterministic, each time you run the same task it

produces the same output. A reversible computation is one that can be run either forward or backward. This means the input can be created from the output. Earlier we considered the division problem  $58432 \div 83$ . A computation that solves this problem and produces the answer, 704, and nothing else, is not reversible. There is no way a computer, or you for that matter, can reconstruct a particular problem if all you know is the number 704. To make this computation reversible, the answer would need to include sufficient information to reconstruct one particular way to calculate it. Something like “704 is the answer to a division by 83” would permit reconstruction of the original problem by an appropriate algorithm.

A probabilistic computation yields a different output each time it is run. At least one of its rules specifies a probabilistic—that is, random—action. Probabilistic computations are neither reversible nor predictably repeatable. They are, however, capable of outputting almost anything you might want, if—and it is a big if—you are willing to run the computation enough times. Modern digital computers are deterministic because designers have gone to great lengths to make them that way. A computer that provides a different answer each time it is given the same input is not very useful for most applications.

Digital computers are possible because physical devices can be built that carry out logical operations based on sequential combinations of yes/no, on/off, or high voltage/low voltage alternatives. Within computers, device states (high and low voltage or on/off switch combinations) are determined by previous device states in a chain of dependent states that begins with the input and initial device states. Since numbers can be represented as states by assigning the value of 1 to the high voltage state and the value of 0 to a low voltage state (or vice versa), computers are able to perform mathematical operations by logical manipulation of sequences of high and low voltage states. Each step in a manipulation is one of those simple mindless steps identified as fundamental to all computational operations by Alan Turing. The yes/no choices are made by circuit structures called logic gates.

Examples of two logic gates and a simple logic circuit made from them are shown in figure 2.2. The circuit carries out simple logic actions by using high or low voltages to represent 1s and 0s. The top symbol represents an XOR (exclusive-or) gate and the middle one an AND

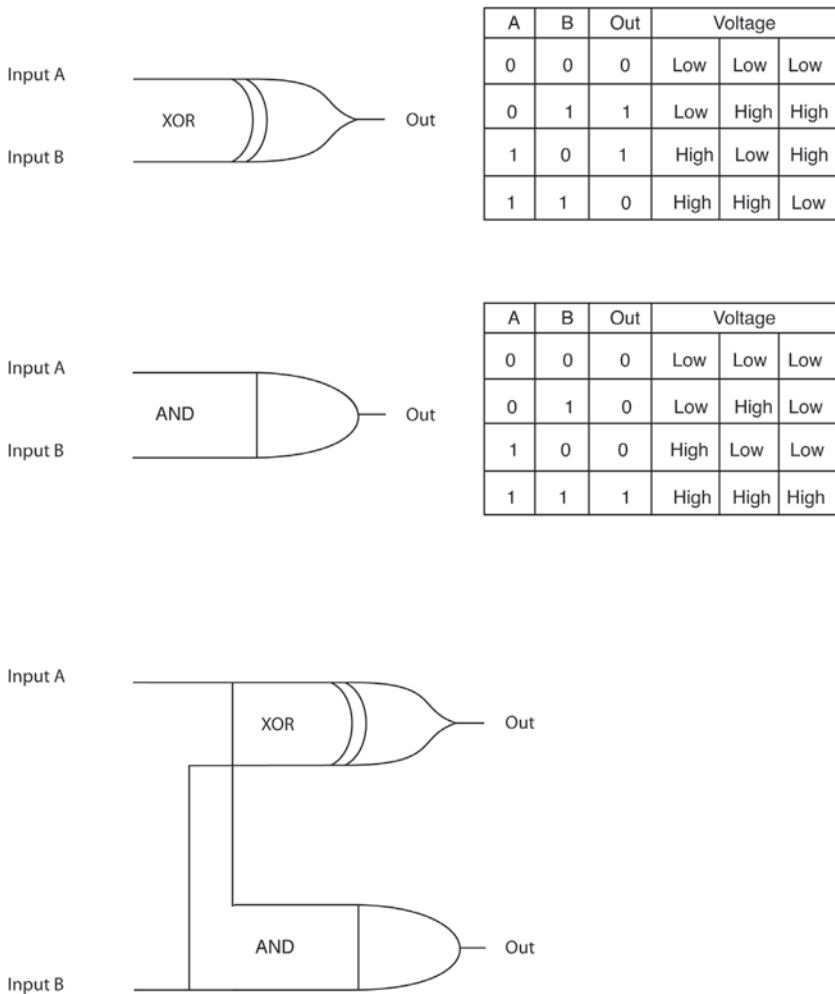


Figure 2.2. Upper and middle diagrams: XOR and AND gates with tables showing all possible inputs and corresponding outputs. Lower diagram: A two bit adder. Inputs A and B may have values of 1 or 0. The two outputs constitute a binary number (in this case 00, 10, or 01) depending on the inputs.

gate. Both gates accept high or low voltages on input wires (labeled A and B in the figure) and output either a high or a low voltage on the wire labeled out. AND gates output a 1 (high voltage) if *both* inputs are high voltage, otherwise they output a 0 (low voltage). XOR gates output a 1 when *exactly one* of their two inputs is a 1, and output a 0



**Table 2.1**

The correspondence between binary and decimal numbers.

Binary	Decimal
00	0
01	1
10	2
11	3

when both inputs are 1 or both are 0. When connected together as shown in the lower diagram, the circuit correctly adds any combination of two 1s and 0s ( $0 + 0$ ,  $0 + 1$ ,  $1 + 0$ , and  $1 + 1$ ). Thus, if input A is a 1 and input B is a 0, upper output is 1, and lower output is 0. The binary representation of decimal number 1 is 01 (see table 2.1). If both inputs are 1, the output will be 10 (decimal 2). This simple adder cannot output 11 (decimal 3); it can only add up to 2. The “rules” that dictate the function of the adder are implicit in the physical structure of the circuit. More complicated logic circuits add, subtract, multiply, and divide numbers of any size.<sup>3</sup>

Computers can also be built so the logical operations they carry out are not hard wired, but are programmable. This means that the XOR and AND gates shown in figure 2.2 would not always be paired with each other in the way shown but instead could be combined in other ways and with other gates to accomplish different logical operations. In such machines, the program determines the overall circuit logic.

When the input determines both the logic and the information to be manipulated, it is customary to divide the input into two parts: program and data. The program part can be thought of as rules that augment a (usually small) rule set built into the machine. During a computation, data is manipulated by following both program rules and built-in rules. Modern digital computers can be programmed to carry out essentially any logical task. Machines with this property are called universal. Universal computers can be programmed to perform any computation that any other computer can perform. Universality defines a very broad class of manipulations collectively known as the computable mathematical

functions. These include nearly all of known mathematics. This is why electronic digital computers are useful for doing so many things.

## How is information defined and measured in computer science?

When considering deterministic computations, any information present in the output must somehow be present in the input—it has to come from somewhere. Electronic computers do not make up information; they rearrange, calculate, copy, and delete, but they don't create it out of thin air. The old folk axiom “there is nothing new under the sun” reflects the spirit of deterministic computation. As discussed in chapter 1, the simplest measure of the information capacity of a string of symbols is its length. This simple measure, however, does not reflect the commonsense notion that information cannot be made up during a deterministic computation. Clearly many computations produce outputs that are longer than their inputs.

A measure of information that is widely used by computer scientists begins with the observation that there are many ways to compute any particular output. Actually, there are an infinite number of ways. If you don't believe this, consider how many different arithmetic problems have the answer 8. The equations:  $4 + 4 = 8$ ,  $2 + 6 = 8$ ,  $12 - 4 = 8$ , and  $440 - 432 = 8$  are just a few; the list is endless. But, for every given output (answer), and language, there is at least one input that has the property that no shorter input exists that produces the desired output. There may be several shortest inputs, all the same length, but there is always some minimum length below which there are no inputs that yield the output.

To convince yourself of this, consider the example used above, namely all the arithmetic problems that have the answer 8. Clearly,  $4 + 4$  is shorter (three symbols) than  $12 - 4$  (four symbols) or  $440 - 432$  (seven symbols). There are ten simple arithmetic operations requiring three symbols in standard English that output the number 8. These are:  $0 + 8$ ,  $1 + 7$ ,  $2 + 6$ ,  $3 + 5$ ,  $4 + 4$ ,  $8 - 0$ ,  $9 - 1$ ,  $1 \times 8$ ,  $2 \times 4$ , and  $8 \div 1$ . They are all the same length, and there are none shorter. When coded in binary, the lengths of these ten expressions are not all the same, but the principle still holds: *for every output, computer and*

*language, there is some input length below which no input yields that particular output.*

In the late 1960s the Russian mathematician Andrei Kolmogorov formalized this observation and used it as the basis for a definition of information different from the Shannon measure. This measure, which is variously called algorithmic information or Kolmogorov complexity, relies on the notion of a universal computer. Recall that a universal computer is one that can be programmed, and because of this, it can perform any computation that can be performed by any other computer. Your desktop computer is a universal computer. Most pocket calculators are not.

Informally, the algorithmic information content of a binary object (a particular string of 0s and 1s) is the length of the shortest input to a particular universal computer that outputs the object. The length of this input will differ somewhat depending on what computer and language you choose, but Kolmogorov provided mathematical proof that the difference in length is at most a constant number that depends only on the computer used and is independent of the output. This means that for very long outputs (vastly larger than the constant), the algorithmic information content is essentially the same no matter what universal computer (and a language) is chosen. If we select a machine that accepts English commands, the input program “repeat 01 1000 times” will result in output consisting of two thousand alternating 0s and 1s (2,000 bits long). This program if converted directly into binary would require about 50 zeros and ones (50 bits) to write. From this we may conclude that the information content of those 2,000 output bits is no more than 50 bits of algorithmic information. By choosing a computer language other than English, this upper limit can be reduced considerably, but never to fewer than about 15 bits because the repeating unit (01), the command “repeat,” and the number 1,000 all contribute to the algorithmic information content.

Algorithmic information shares with Shannon information the property that it cannot be created during a deterministic computation. The information content of the output can be less than that of the input, but not greater. Thus, algorithmic information conforms with our intuitive notion that information cannot be created out of thin air.

If we subtract the algorithmic information characterizing an object from its length, the difference is called the “redundancy” or

“compressibility” of the information. This represents the amount that a clever representation of an object can be smaller (shorter) than the object itself without losing the ability to recreate it by a suitable deterministic computation. “ZIP” files are smaller representations of larger files. “Unzipping” them is to perform a computation that recreates the original longer pattern of zeros and ones.

The idea of randomness is tricky to define with mathematical precision, and algorithmic information provides a powerful definition. Whenever the algorithmic information and the length of a sequence of symbols are equal (actually, within a small constant determined by the computer used), the structure of that sequence is considered algorithmically random. This means that no amount of deterministic computation can produce a shorter sequence of symbols without irreversibly losing some aspect of what is being represented. Sequences with this property have no redundancy and are said to be randomly structured.

This definition of randomness reveals a curious feature of the algorithmic information concept. When comparing two sequences that are the same length, the one that is most nearly random has the highest information content. When we consider complicated things, we do not usually think of them as random but rather as having intricately specified structure. To glimpse the relationship between randomness and complexity, consider that to reproduce a particular sequence requires that every element be precisely specified. The sequence 01010101 . . . with one thousand 01 repeats is an object with a lot of internal repetition. Because of this, a short program can be written that outputs the larger object. Random things lack repetition, and any program that outputs a particular random sequence must somehow include the sequence itself. Complex things are characterized by a lot of detail, and all that detail often requires long programs to generate, so complex things tend to have a high algorithmic information measure, but this is not always so.

### Can great complexity be created by short (i.e., simple) programs?

Mathematical analyses of logical operations reveal fundamental limitations on what deterministic computations can and cannot do. The following three principles are important:

Principle #1: Simple inputs and simple transformation rules *can* (but don't necessarily) lead to outputs with great complexity.

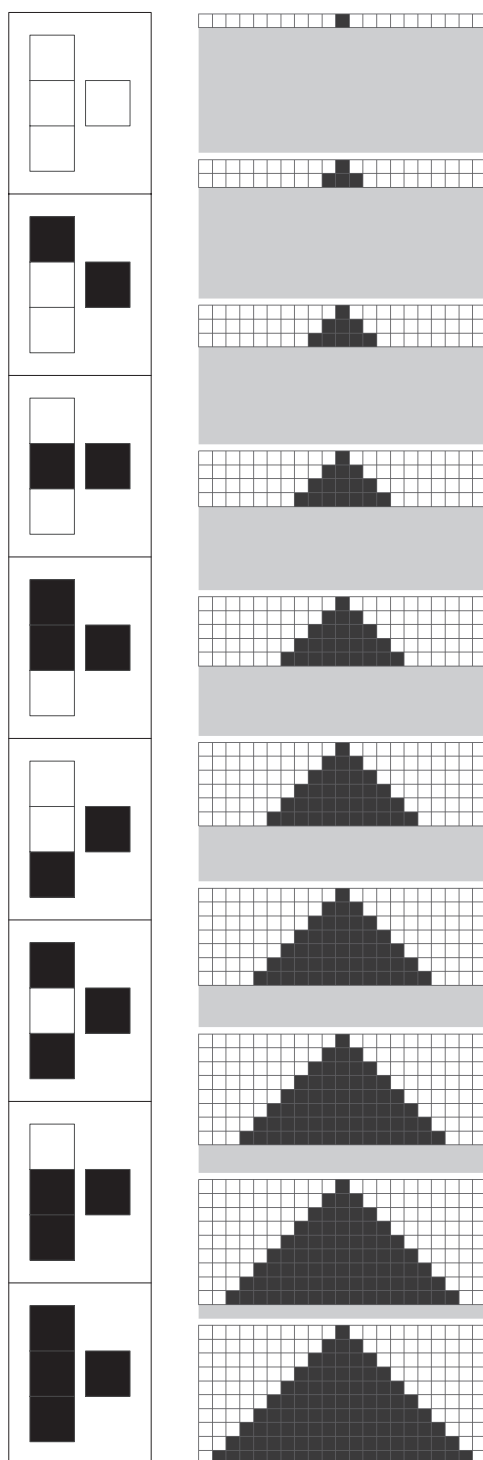
Principle #2: When great complexity is created by applying simple rules to simple input, many computational steps are required for the transformation.

Principle #3: Creation of many complex objects requires complex programs. Such objects cannot be created by the application of simple rules to simple data no matter how many computational steps are devoted to the task.

Stephen Wolfram spent years exploring principle #1 and has made many of his findings available in *A New Kind of Science*.<sup>4</sup> He defines NKS (New Kind of Science) systems to be logical systems in which simple rules and inputs lead over time to great apparent complexity and argues that much of the complexity we observe in the physical universe illustrates principle #1 in action.

Figures 2.3 through 2.6 demonstrate four NKS systems. In these examples, states of a device known as a cellular automaton (CA) follow a few simple rules and change over time measured in steps. Many NKS systems are not based on CAs, but this class of computations serves well to illustrate the principle. CAs are one-, two-, or three-dimensional arrays of cells (not to be confused with biological cells) whose properties are periodically updated. In figure 2.3, the rule set has eight rules shown graphically in the left panel. Each rule is depicted as four boxes within a larger box. The rules shown in the left panel are rotated 90 degrees relative to the computation shown in the right panel. The first rule in the set is interpreted to mean that any time three cells in a row are all white the cell centered below them in the next row is made white. With the second rule, if the rightmost of three adjacent cells is black, the cell centered in the next row below is made black, and so forth. Looking at all eight rules, it is apparent that a cell in a new row will be black when any of the three cells above it are black and white only when none of the three cells immediately above it are black.

The way to turn the rule set shown into a computation is to start with a row of cells. You can start with any pattern of black and white cells. In the right panel of figure 2.3 the initial row is all white except for one black cell (step 1). Next, create a row below the first by applying rule set 254. The result is an all-white row with three black cells (step 2).



*Figure 2.3.* A cellular automaton employing rule 254 (left panel) produces an ever growing black triangle (right panel).  
Copyright Wolfram Research, Inc. (see note 4).

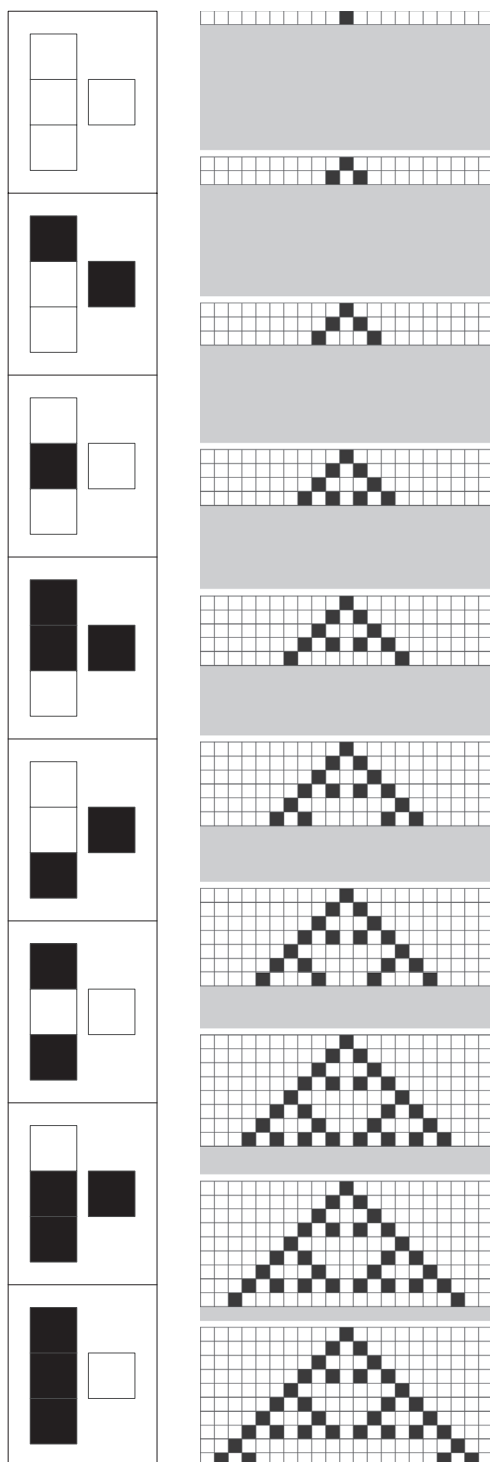
Following this, a third row is created based on the pattern of cells in row two (step 3). This row has five black cells. The computation may be continued indefinitely. As one might expect, given the simple nature of the input data (row 1) and the simple rules being applied, the result is also simple, namely a growing black triangle. Rule set 254 can never produce patterns significantly more complicated than the pattern present in the input row. Figure 2.4 shows a more interesting CA that implements a different rule set. This example also begins with a single black cell in an otherwise all-white initial row. The pattern produced is known as the Sierpinski gasket, a classic fractal pattern in mathematics. As updating continues, larger and larger triangles appear together with smaller triangles of all sizes.

When cell properties are limited to two colors (black or white) and the color of each cell is determined by the colors of three adjacent cells above (as is the case for figures 2.3 and 2.4), eight rules are required to cover all possibilities and there are 256 possible rule sets. Most produce simple patterns or no pattern at all after the first row. A few yield more complex patterns.

Figure 2.5 shows a pattern with both regular and irregular elements resulting from rule 110 acting on an input row with a single black cell. Remarkably, rule 110 is universal. This means that a universal computer can be constructed by applying rule 110 over and over to a suitable input row. This theoretical result proves that universal computation can be logically simple and that simple rules are capable of producing great complexity.

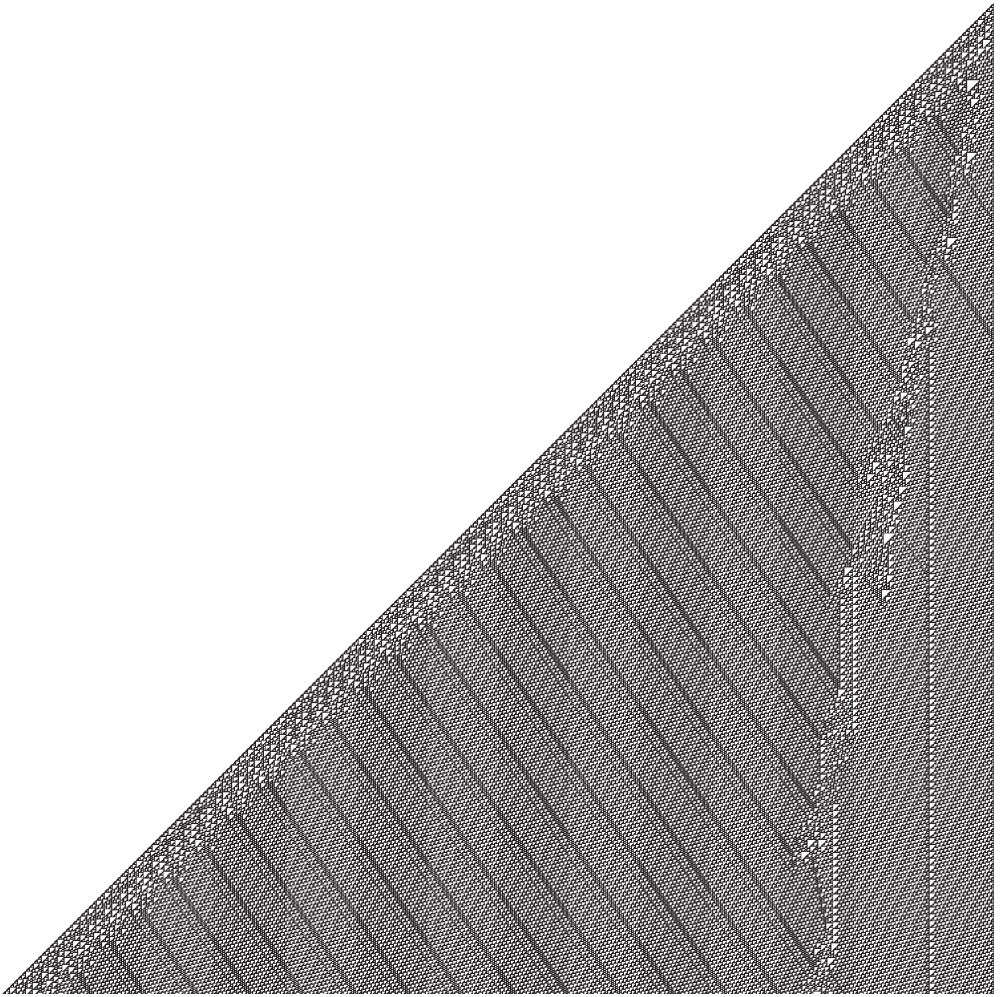
By increasing the number of cell colors from two to three (white, black, and gray), the number of rules needed for completeness increases from eight to twenty-seven and the number of possible rule sets increases to over seven million; but once again, a small number of these rule sets produce complex patterns, while most yield only simple output.

A slightly different scheme allows shades of gray and determines the color of each cell by averaging the color of the three adjacent cells above it. With this scheme, seven rules are needed if cells can take on one of seven shades: white (3w), very light gray (2w + g), light gray (w + 2g or 2w + b), gray (3g or w + g + b), dark gray (2g + b or w + 2b), very dark gray (g + 2b), and black (3b). There are 2,187 ways (rule sets) for assigning shades of gray to cells in this way. Once again most CAs employing such rule sets produce simple outputs, but a few result in wildly creative patterns. The result of rule set 1635 acting on an initial



*Figure 2.4.* A cellular automaton implementing the first 100 steps of rule 90 (shown to the left). The pattern is known as the Sierpinski gasket. Copyright Wolfram Research, Inc. (see note 4).

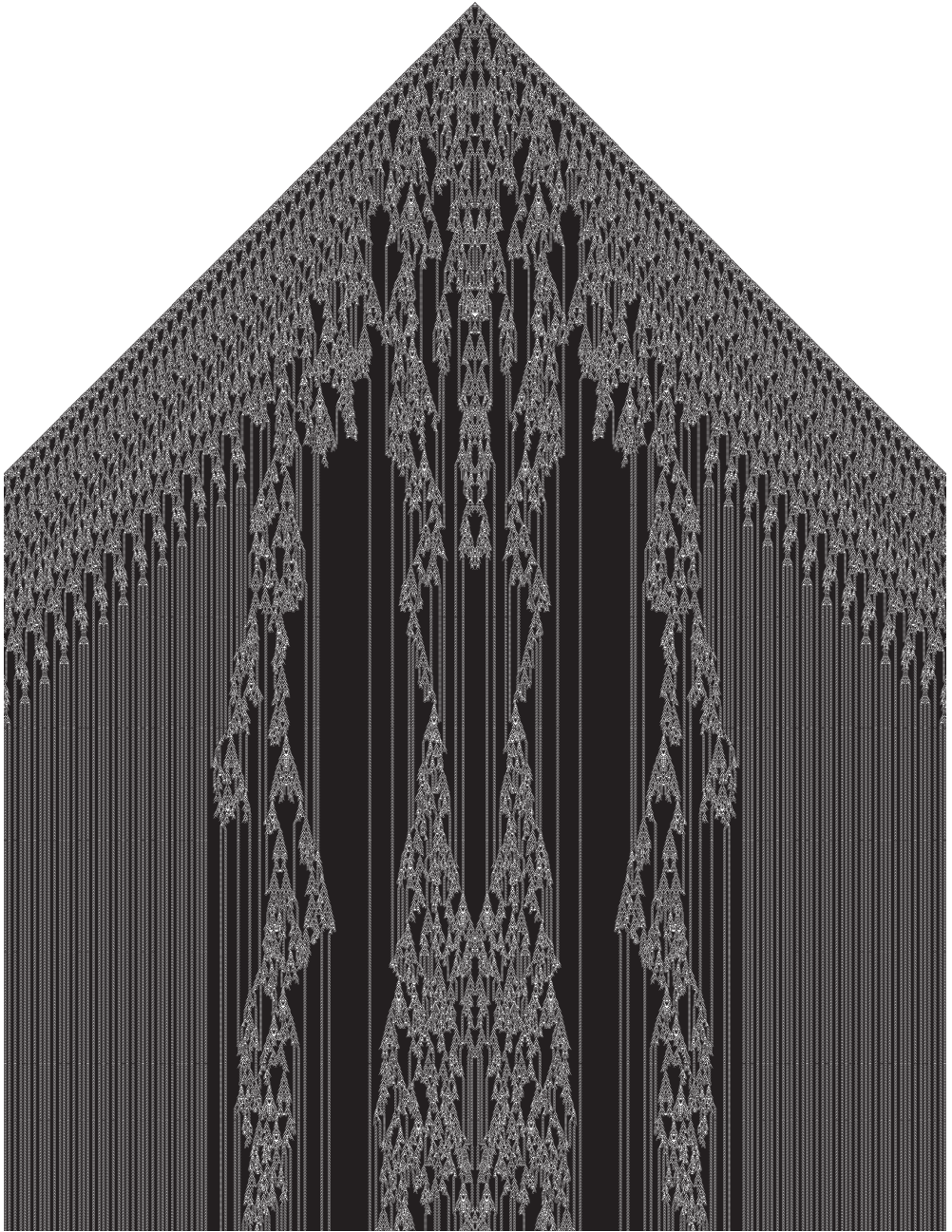




*Figure 2.5.* The first 1000 steps of the left half of the pattern created by implementing rule 110 (not shown). Copyright Wolfram Research, Inc. (see note 4).

row consisting of one black cell surrounded by all white cells is shown in figure 2.6.

These four examples of simple CA computations give the flavor of what can happen when very simple input (in this case, a single black cell in a field of white cells) is repeatedly acted upon by simple rules. Clearly the answer to the question posed in the title of this section is yes, some simple rules acting on simple input can generate significant



*Figure 2.6.* The pattern revealed implementing the first 3,000 steps of rule 1635 (not shown) employing three colors (shades of gray) and an averaging rule. Copyright Wolfram Research, Inc. (see note 4).

output complexity. There are a wide variety of systems in which computer scientists have tested simple rules repeatedly applied to simple input, and the general conclusion drawn is that combinations of simple rules and inputs that lead to significant apparent complexity are widespread but not common, and even within those systems that can create significant complexity, most rule sets do not.<sup>5</sup>

Figure 2.6 illustrates how uninformative the several information measures we have discussed are when compared to our everyday notion of complexity. Everyone would agree that the pattern in the figure is complex. A single row of cells near the bottom of the pattern shown in figure 2.6 would seem to encode a lot of information because it is complicated and long. But, we know the row is characterized by low algorithmic and Shannon information measures because the rule set and input data (a single black cell) that generated it are simple. Thus, the information measures we have discussed are not reliable indicators of complexity when judged by a human observer.

The behaviors of NKS systems are usually explored by using a computer, but the four examples we have seen could be implemented instead by arranging black, white, and gray cards on a large table. Next, we will look at a system that does something very similar but in a test tube. It can be thought of as carrying out computations through the self-assembly of small units called molecular tiles. The outputs of these computations are physical structures.

## Can physical structure be computed?

The modern scientific conception of our universe is one in which a small number of relatively simple rules acting on simple initial conditions have resulted over eons of time in vast complexities. Physical systems are constrained by rules as much as computational systems, but the rules are the laws of chemistry and physics rather than of human design. Were physical laws not logical, mathematics would be of no use for characterizing nature, and there would be no scientific field of physics. The parallel roles of rules in computation and physical laws in physical process have stimulated a long-standing debate over whether or not it is legitimate to consider the universe to be a computer. This section does not attempt to answer that question but does illustrate

the impossibility of drawing a conceptual line between physical process and computation.

Erik Winfree and colleagues at Caltech have created physical systems whose behaviors are similar to one-dimensional CAs.<sup>6</sup> These are based on chemical structures called “molecular tiles.” Tiles are flat structures that have two or more edges that can bind to other tiles. Appropriately designed tiles “self-assemble” into predetermined structures.

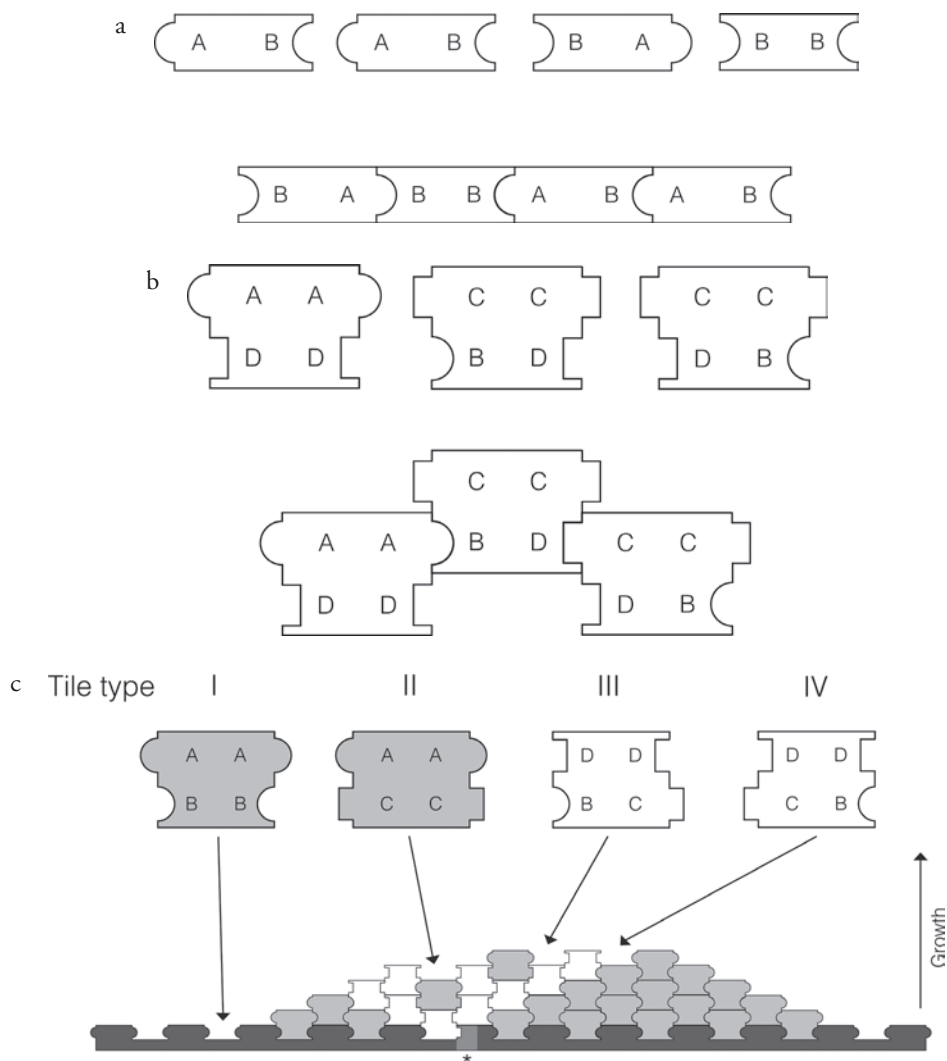
Flat tiles that allow binding only at their edges form one- and two-dimensional structures, and three-dimensional tiles are required to form three-dimensional structures. In figure 2.7a the top row shows four flat tiles.<sup>7</sup> Each tile has an A or B binding site on each end. If we invoke two rules—A binds only to B and A does not bind to upside-down B (and B does not bind to upside-down A)—then the only way these particular four tiles can bind together in a single structure is shown in the lower row.

Figure 2.7b illustrates three tiles with four interaction sites located on the ends. In this example there are four interaction site types: A, B, C, and D. The basic rule is that A binds to B and C binds to D. Again, upside-down binding is not allowed. In these tiles, each lettered binding site is associated with a particular shape. The particular combination of binding sites shown allows staggered configurations to form as shown in the structure below the three separate tiles.

In figure 2.7c the four tile types shown are used to build a growing structure. Only complementary shapes are allowed to bind. If we employ a rule that tiles permanently bind to other tiles only when *two sites* on each tile fit together, then the tiles shown in the upper panel of figure 2.7c cannot initiate any permanent structure because end tiles can only bind to one other tile.

To get around this problem, it is necessary to start the assembly process with a template or seed row. The growing structure shown in the lower panel of figure 2.7c initiates from a template row shown in dark with a single gray binding site in the middle. Type I tiles can readily bind to the template everywhere except at the gray binding site marked with an asterisk. At this site only tile type III can bind. Assembly of the first row of tiles bound to the template permits a second row to form above it. The second row also consists mostly of type I tiles but with type III and type IV tiles directly adjacent to the single type III tile in row one. The second row enables a third and so





*Figure 2.7.* a. Diagram of self-assembling tiles each having a binding site on opposite ends and a structure they can form. b. Self-assembling tiles with four binding sites. This particular combination of sites allows only staggered configurations as shown on the right. c. Assembly of four tile types on a template (dark row). In the first row above the template, tile type I can bind everywhere except for one spot (marked with an asterisk). Tile type III binds at this place. Each successive row presents novel features (see note 7).

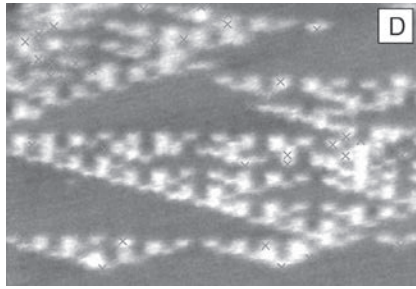


Figure 2.8. Triangle patterns formed by tiles made of short bits of DNA and visualized using an atomic force microscope. White tiles are designed to protrude above gray tiles resulting in the color contrast. Errors are marked with an  $x$  (see note 7).

on. Each successive row enables another row to form above it and is exactly specified by the pattern of tiles in the row below. When type I and type II tiles are shaded gray (as shown) and type III and IV tiles white, the growing pattern of white tiles reveals the Sierpinski gasket. This is the same pattern computed by CA rule 90 in figure 2.4. The template row with a single type III binding site plays the same role in the physical tiling system as the single black cell embedded in a row of white cells in one-dimensional CA structures shown in figures 2.3 through 2.6.

Figure 2.7c is a drawing. Figure 2.8 shows a microscopic physical Sierpinski gasket that formed spontaneously in a test tube from tiles made of short pieces of DNA. These physical tiles are chemical versions of the tile designs shown in figure 2.7c. The structures seen in figure 2.8 formed when the DNA tiles were mixed together with a small amount of template in an appropriate salt solution. In principle, molecular tiling systems with appropriately designed tiles and templates are capable of creating rows of tiles with any desired sequence. This means that, theoretically, any algorithm that can be implemented in an electronic computer can be implemented by assembly of an appropriate tile set and template. It also means that any conceivable finite two-dimensional structure can be created by self-assembly if one has the appropriate tile set and template. Put more generally, *“matter can be programmed to assemble itself from very simple components into useful structures.”*<sup>8</sup>

A demonstration of the potential computational power of tile assembly was provided by Eric Winfree in his PhD thesis where he showed

that tile sets could be designed that would carry out (universal) Turing machine computations. This proved, in principle, that anything that can be computed by an electronic digital computer can also be computed with molecular tiles.

In practice there are severe limitations to the tiling systems demonstrated to date because of their high error rates. Molecular systems do not behave as neatly as electronic computers because molecules sometimes bind incorrectly. In particular, the rule that two sites must be bound in order for a tile to become incorporated into a larger structure is hard to enforce. A number of error events occurred in the experiment recorded in figure 2.8. In simple self-assembly systems, errors propagate destroying the deterministic nature of the assembly and limiting the design complexity of structures that can be successfully formed. We know, however, that it is possible for very complex structures to form through self-assembly. After all, living organisms are such structures.

The problem of error propagation is not unique to molecular tile assembly systems; it actually characterizes all forms of deterministic computation. Your home computer gets around the problem by reducing the number of errors to an extremely low level and, sometimes, by incorporating error-correcting strategies. Error correction refers to mechanisms that check whether or not the last action was done correctly before proceeding on to the next step. In molecular scale processes, thermal noise is unavoidable. So, any practical implementation of computing by molecular assembly must employ error correction. Living systems that depend for their very existence on the molecular self-assembly of complicated structures exhibit many such strategies. Most of these have yet to be effectively implemented in artificial test tube assembly systems.

Self-assembly of physical objects is characterized by having computation rules physically embodied in tile designs. This is illustrated by comparing figure 2.7c with figure 2.9. In figure 2.9 tiles appear as staggered rows of rectangles labeled with 1 or 0. In figure 2.7c tiles form rows based on their complementary shapes. Figure 2.9 reveals the computational rule: “add a 1-tile whenever one of the two tiles below it is a 1, and add a 0-tile whenever the tiles below are both 0 or both 1.” This is the “exclusive or” (XOR) logic function implemented as a logic gate by the device shown in figure 2.2. The Sierpinski gasket results from the repeated application of the XOR logic function

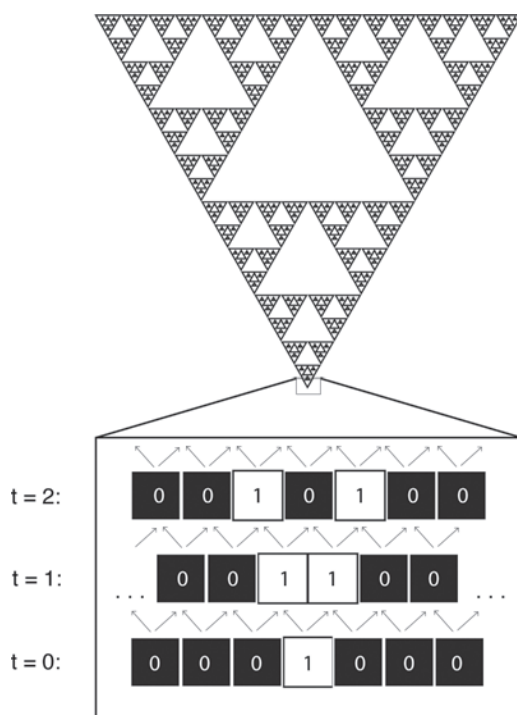


Figure 2.9. Logic behind the formation of the Sierpinski gasket using tiles.  $t = 0$ ,  $t = 1$ , and  $t = 2$  indicate successive rows, each formed according to the pattern of the row below (see note 7).

starting from a single 1. Rule set 90 (figure 2.4) is another implementation of the XOR logic function. Construction of the molecular structures shown in figure 2.8 makes chemical the repeated application of this same XOR logic function. The rule is not written down anywhere, but is implicit in the design of the tiles used. Different tile designs can implement other logic functions and cause other structures to form.

Figure 2.9 also helps us understand the role of the seed sequence. An expanding array of Sierpinski triangles results when the XOR function is applied to the situation in which the initial row is all 0s containing a single 1. If we imagine a seed row consisting of widely separated 1s, the result would be Sierpinski triangle patterns initiating from each 1. Eventually, the triangle patterns collide and the resulting interference

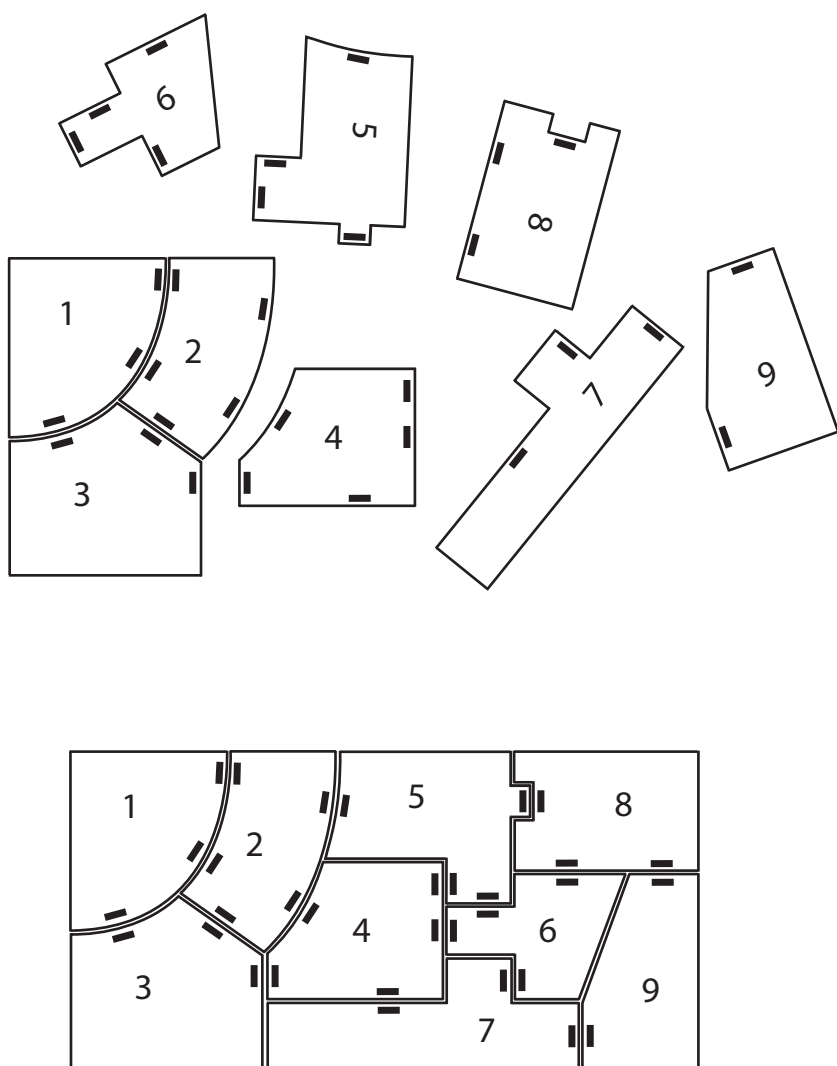


patterns get really complicated. An initial seed sequence consisting of a complicated pattern of 0s and 1s can cause an equally complex pattern of 0s and 1s to be created in subsequent rows. The clever use of seed sequences allows the computation of specific complicated rows using nothing more than the XOR function as a rule. But, the XOR rule is not enough to achieve universal computation. For that you need a special (and rare) rule set like rule 110 (figure 2.5).

To imagine tile self-assembly at the scale of human activity, consider a jigsaw puzzle for which each piece has been modified by adding small magnets in strategic locations along the edges. The strength of these magnets is such that two puzzle pieces will only bind tightly together when two magnet pairs are correctly aligned. This rather special puzzle has the almost magical property that to assemble it one needs only to place the pieces on a flat surface, like a cookie sheet, with all the faces up, and shake the sheet horizontally so that the pieces jiggle around this way and that but don't flip over. Eventually, two at a time, the pieces will meet at appropriate orientations and snap together. Slowly the puzzle will self-assemble. Figure 2.10 shows such a puzzle.

In figure 2.10 thick black regions are magnets, and the vigor of shaking is set so that single magnet pairs cannot remain permanently bound together; only when two or more magnets come into contact at the same time do the puzzle pieces remain stuck. Only pieces 1 and 2 meet the condition that they can bind together without help from other pieces. Once 1 and 2 have bound together, piece 3 can bind to them; after 3 is bound, 4 can bind, then 5, then 6. Pieces 7 and 8 can bind after 6 in any order, but only after both 7 and 8 have joined the complex can piece 9 bind, finishing the puzzle.

Tile self-assembly bridges the conceptual divide that normally separates computation from physical process. Ultimately, both physical structure formation and computation can be understood in terms of configurations interacting with other configurations over time. Deterministic digital computation constitutes an extreme where configurations are formed from only two kinds of parts (0s and 1s) and employ rules that are always obeyed (no errors). Physical systems also compute, but they usually have more kinds of parts and are noisy, meaning that mistakes are common and precise outcomes are often not predictable in detail even when the rules are known.



*Figure 2.10.* A self-assembling puzzle consisting of nine pieces. The small black marks are magnets. To become included in the assembly each piece must bind to another piece with at least two magnets. The upper panel shows the first three pieces assembled. The lower panel shows the completed puzzle.

## Why have long programs?

When most of us think of computations, we don't imagine self-assembling tiles or NKS systems or even physical systems based on simple rules; rather, we envision applications requiring long programs such as word processors and spreadsheets. Most of the useful computer applications in our experience require long programs. By considering all possible outputs of every conceivable computation, mathematicians have proven that the vast majority require long inputs to produce. Given the close relationship between computation and physical assembly, it stands to reason that many things we might want to make would require long programs. For NKS and tiling systems with a finite number of rules or tiles types, this means that most possible rows (outputs) cannot be achieved without a complicated seed row. Thus, despite the complexity of CA rows created by rules 90, 110, and 1635 (figures 2.4, 2.5, and 2.6), the vast majority of conceivable rows will never appear in any of these computations as long as the seed row is simple. Most times, to get a particular row you might want, you need to start with a carefully designed and already complicated first row. The details of this row determine the outcome for any given rule set. Repeating for emphasis what was previously stated, ***computer theory tells us that most conceivable outcomes simply cannot be created by a deterministic process if the input (rule and seed) is simple.***

On the bright side, it seems there is nothing within the scope of symbolic languages that appropriate long programs cannot produce. Those who regularly use desktop computers have experienced this. Word processors, spreadsheets, Web page editing programs, and computer games are just a few examples of long programs that display a dazzling spectrum of outputs. Most of these everyday computer applications are simply not possible with short programs, no matter how cleverly designed.

This same principle almost certainly applies to the physical world. Consider the formation of living organisms and many products of human technology. Long, complex instructions are required for the creation of both organisms and automobiles. From the principles of computer science we can guess that such things simply are not possible without employing extra information to supplement the laws of chemistry and physics. This is the role of instructions.

To summarize the many concepts presented in this chapter, a computation is any process that can be characterized in terms of logical rules acting on preexisting configurations to produce new configurations. It doesn't matter if these configurations occur inside computer circuitry, in a test tube containing molecular tiles, in a living cell, or in a self-assembling jigsaw puzzle.

Configurations are states of systems, and when states can be distinguished from other states, a system carries information. Just as important, system states can represent other system states. This offers the opportunity for meaning to be transferred from one system to another. The terms *information* and *meaning* get tangled up in normal word usage. In chapters 1 and 2 we encountered three ways to define and measure information. Length is simple, but doesn't tell the whole story; Shannon information is based on probabilities of states; and algorithmic information is defined by process (a computation that produces the object). Length can increase in a deterministic computation while algorithmic information and Shannon information cannot. All three measures can decrease as the result of a computation. No one has devised a quantitative measure of meaning.

An important principle of computation is that significant complexity can be created by simple rules acting repetitively on simple initial conditions. NKS and DNA tiling systems provide visual examples of this, and so do physical structures that form through repeated application of the laws of chemistry and physics to simple initial configurations of matter. In the next chapter we will see a few examples of this kind of structure.

Another important principle of computer science is that some complicated things cannot be computed by using simple input and rules. Life and many products of life, including human societies and human technology, seem to require long programs. Long instructions play the same role in the physical world that long programs play in the computational world. They allow the creation of complicated objects and behaviors that are not possible without them. We will see some examples of this in chapter 4.

# 3

---

## Structure for Free

If physical processes compute, what and where are the programs?

Imagine a water molecule floating around in the atmosphere, freely moving this way and that until by pure happenstance it encounters a growing snowflake. Depending on the angle of approach, it sticks like a fly to flypaper, or it bounces off, continuing on its way. This seemingly haphazard process when repeated over and over builds a beautiful and regular structure. Think of a river flowing over a mossy rock. Water molecules in uncountable numbers interact, this time to form eddies spinning their way down the stream. Now imagine the grand sweep of a vastly large but wispy cloud of hydrogen gas in the emptiness of space slowly coalescing through gravitational self-attraction to form a rotating galaxy complete with billions of stars and planets. Finally, think of those DNA tiles discussed in the last chapter that by the simple act of locking together according to their chemical shapes assemble a regular pattern of triangles.

What do these processes have in common? All are spontaneous. All have basic units of some sort: atoms, molecules, or molecular tiles

that interact according to simple rules, and none require an outside intelligence in order for the structure to form. There is always some rule of attraction at work. Once formed, structure interacts with other structure resulting in new outcomes. Structure builds on structure, with each achievement serving as the seed for some future formation. If you could follow a chain of events back in time, structure preceding structure, eventually the building blocks themselves would emerge as a consequence of the rules in play. The start of it all, we are told by astronomers, was 13.7 billion years ago. The late British cosmologist Fred Hoyle coined the term *big bang* during a radio program in which he contrasted “steady state” theories (which he favored) with theories featuring a definite beginning (now favored by nearly all cosmologists). In big bang theories, the entire universe started out infinitesimally small, infinitely dense, and infinitely hot; physicists call it a singularity.

This is the prevailing scientific view of our universe. The laws of nature are pervasive; they operate at all scales and at every time and place. Time, space, and perhaps the rules themselves began with the big bang. The laws are not written anywhere in nature, but they can be discovered and described by human investigators.

The laws of physics are mostly mathematical, and they allow calculation of how the universe must have proceeded from when it was very small and hot to what we have now, but the laws do not explain their own origin. The Holy Grail of modern cosmology is to come up with a self-contained theory that explains the laws. Readers interested in this issue can consult a brief reading list provided in the notes.<sup>1</sup> Fortunately, we can learn the laws of physics and study their consequences without having such a “theory of everything”; they work whether or not we understand them at a deeper level.

Some of the rules are deterministic, that is, they always result in the same outcomes when presented with the same situation. Others are stochastic, meaning they lead to different outcomes even if presented with the same situation. We also know there are situations in which deterministic laws interact to produce chaotic activities whose precise outcomes cannot be calculated beyond the near-term future. Because of stochastic and chaotic happenings, precise future details sometimes cannot be predicted; thus, each snowflake is unique even though the rules are the same.

In chapter 2 we saw structures forming in simple computations. These examples show that very complicated patterns can sometimes arise when simple rules operate on simple input. When studied systematically, it turns out that many, probably most, combinations of simple rules and simple inputs do not produce complicated structure.<sup>2</sup> Many combinations produce nothing, some result in simple repetitive patterns, while rare combinations produce complex patterns. In physics and chemistry also, laws acting on simple or unexceptional initial configurations may result in nothing, or less commonly in repetitive structures, and even less commonly in complex structures. Understanding how a particular structure came into being requires knowledge of both the relevant laws and the initial conditions.

Like computer outputs, physical structure formation is indeterminate in the sense that there are many ways a particular object might come into being. It may be difficult to discover which of the possible ways actually produced an object under study. Fortunately, understanding a structure usually does not require knowing exactly how it formed; it is enough to know how it might reasonably have come into being. Consider again a snowflake. Water molecules add to the growing structure at many different locations on its surface and approach from many angles; the details of each molecular collision determine snowflake uniqueness, but we need not know them for us to have a general understanding of how snowflakes form. The principles of snowflake formation are pretty much the same snowflake to snowflake even though the molecular details of each individual formation are different.

When considering possible formation scenarios for how something might have formed, one must entertain the formal possibility of random creation, the haphazard throwing together of parts. Structures can and do occur by chance, but the probability of particular configurations occurring by chance becomes smaller and smaller as the number of parts increases. For objects with even a modest number of parts, the probability of one particular ordered structure occurring randomly can be so small that it is never expected to happen in the entire lifetime of the universe. I call this the *probability problem*. In situations where the probability problem is encountered, another explanation is required.

Chapter 1 discussed dividing things into two types of structures: Type I (based on simple rules) and Type II (requiring complicated

rules). A completely different way of categorizing structures emerges from the study of thermodynamics. In this way of dividing things up, some kinds of objects form when systems are near to thermal or chemical equilibrium, and other kinds of things form when systems are far from thermal or chemical equilibrium. Examples of near-equilibrium structures are crystals and solar systems. Such structures persist until an outside force changes the system. Structure always results from a particular balance of forces that constitute a system's lowest energy state. Examples of far-from-equilibrium structures are vortexes in a stream, waves in a lake, and our sun. Such structures are referred to as stable states of their system rather than equilibrium states. Formally, living organisms are stable nonequilibrium states. Nonequilibrium structures also owe their existence to a balance of forces, but they only form in situations where energy flows steadily through a system such as when you plug in an appliance or when wind blows over a lake making waves. If the energy flow stops, stability is lost and the structure dissipates. To recap, in equilibrium situations energy is needed only to change structure while in nonequilibrium situations, energy is needed to maintain it.

Another way to divide things is into static structures (those that don't move) and dynamic structures (those that do move). Movement is found in both equilibrium and nonequilibrium structure classes. Thus, a solar system with planets orbiting a sun and moons orbiting planets is an equilibrium structure characterized by movement. The movement does not require continual input of energy from the sun or anywhere else. In contrast, the vortex that forms when you let water out of a bathtub is a [nearly] static nonequilibrium structure even though, invisibly to you, molecules are racing around as they go down the drain. The vortex persists largely unchanged so long as water is running out of the tub. The necessary flow of energy is provided by water moving closer to the center of the earth.

Another complication we encounter is that nature is full of ephemeral structures. These are things that are not long lasting and that constitute transient states of systems. There is no clear distinction between long-lasting structure and transient structure. Our solar system seems quite permanent to us, but over the lifetime of the Milky Way galaxy it is transient. In the realm of nonequilibrium structure, it is common for structures to morph from one form to another without dissipating.



What makes any physical structure possible is the occurrence of balances between opposing forces. In specific situations the forces described by physical laws counter one another. Some laws of physics describe tendencies of matter to clump, and others describe tendencies to dissipate. Given the existence of matter and the fact that many of the mathematical relationships are nonlinear, it is almost inevitable that there would be balance points between attractive and dissipative forces. These balance conditions result in structure. I call it structure for free<sup>3</sup>; the term applies when opportunities for structure are inherent to a system (and its rules). No preplanning is required; structures just form when conditions are right.

We will now look at two examples illustrating how the interplay of attractive and dispersive rules can lead to the spontaneous formation of structure. In both, structure forms as a system approaches thermodynamic equilibrium. No instructions are required.

### How do we explain a grain of salt?

Table salt consists of sodium and chlorine. Like all atoms, sodium and chlorine are made of parts: positively charged protons, negatively charged electrons, and uncharged neutrons. The protons and neutrons are bound tightly together at the center of the atom by the strong nuclear force, while the electrons are located on the outside. One law of our universe is that positive and negative charges attract. This attraction is inversely proportional to the square of the distance between charged objects, a relationship known as Coulomb's law. This law can be stated in the form of a simple algebraic equation:  $F = -k \cdot q_1 \cdot q_2 \div r^2$ , requiring ten symbols to express<sup>4</sup> ( $q_1$  and  $q_2$  are each considered a separate symbol; the dots are a symbol signifying multiplication). This may or may not look complicated to you depending on your previous education, but according to any of the definitions of information we encountered earlier, Coulomb's law doesn't represent very much information; it is a simple rule.

Coulomb's law accurately describes the attraction between a positively charged proton and a negatively charged electron and also the repulsion between two protons or between two electrons. As a consequence of this rule, the most favorable state for most atoms is equal

numbers of electrons and protons bound together as closely as possible. Sodium atoms have eleven protons and eleven electrons, while chlorine has seventeen of each. When atoms have the same number of protons and electrons, we say they are electrically neutral. It is possible, however, for atoms to have different numbers of electrons than protons. In these cases they have a net electrical charge, positive or negative, and we call them ions.

According to Coulomb's law the closer two charges are to each other, the greater the force acting on them. If this were the only law, then the eleven electrons of every sodium atom would crush together with the eleven protons, and there would be no sodium atoms as we know them. This doesn't happen because of quantum mechanics and a rule called the Pauli exclusion principle. Quantum mechanics is a rule that is also possible to write as a simple equation.<sup>5</sup> A feature of the quantum mechanics rule is that systems are partitioned into "energy levels." For atoms this means that electrons are located in energy levels and cannot get closer to the protons than permitted by the lowest energy level. The Pauli exclusion principle states that two identical particles cannot occupy the same energy level. Because electrons have a property called spin (which can have two values), energy levels involving electrons may contain two electrons, each with a different spin. The outcome of the interacting rules is that the lowest energy configuration (the equilibrium configuration) for a sodium atom is two electrons occupying energy level one, two in energy level two, two in energy level three, up to energy level six, which has only one electron. All higher-energy levels have no electrons.

A feature of quantum mechanics when applied to atoms is that those with two, ten, eighteen, thirty-six, fifty-four, and eighty-six electrons are particularly stable. Because sodium has eleven electrons, little energy is required to pull one electron away from an atom (achieving ten). When the eleventh electron is lost, the structure becomes a sodium ion with eleven protons, ten electrons, and a net electrical charge of +1. The situation is similar for chlorine atoms in which two electrons occupy each of the lower-energy levels and the ninth has just a single electron. Because neutral chlorine has seventeen electrons, the extra stability of eighteen electrons means that chlorine readily accepts an extra electron (to make eighteen). This converts the chlorine atom into a chlorine ion with an overall electric charge of -1. The interplay

of quantum mechanics, Coulomb's law, and numbers provides each chemical element with distinctive properties.

When equal numbers of sodium atoms and chlorine atoms are mixed, a spontaneous transformation occurs in which the system transitions to a new equilibrium state. In this chemical reaction heat (energy) is released, and every sodium atom loses one electron, and every chlorine atom gains one electron. The positively charged sodium ions and negatively charged chlorine ions attract and repel one another according to Coulomb's law, but cannot get too close because filled-energy levels (orbitals in chemistry) cannot interpenetrate. The optimal balance between the counteracting rules occurs when a very regular three-dimensional structure forms in which every sodium ion is surrounded by chlorine ions and every chlorine ion by sodium ions. The physical dimensions of the energy levels dictate the optimal distance between ions. Sodium ions and chloride ions are different sizes, and the configuration that best balances the different demands imposed by physics and geometry is a regular lattice with six sodium ions nearest each chloride ion and six chloride ions nearest each sodium ion. This regular cubic structure extends in all directions and is what we call a salt crystal; it forms spontaneously whenever sodium ions and chlorine ions encounter one another within certain temperature and concentration limits. This is especially likely to happen on Earth when sodium and chloride ions are dissolved in water and the water slowly evaporates away.

A salt crystal represents the low-energy (equilibrium) configuration of sodium and chlorine ions in air at room temperature. No flow of energy is required to maintain it, and no external source of energy is needed for the crystal to form. A crystal represents a simple example of structure for free. Three simple rules—Coulomb's law, the Pauli exclusion principle, and quantum mechanics—interact to determine the structure. The formation of a salt crystal can be understood in terms of a reasonably probable starting condition (sodium ions and chloride ions dissolved in water and the gradual removal of the water, for example) and three rules. The presence of other chemicals can complicate the situation and lead to the formation of other structures; but conditions favorable for the formation of salt crystals are not that uncommon, and halite (crystalline sodium chloride) is a common mineral on Earth.

Is crystal formation a computation? After all, we saw in chapter 2 that DNA tiles “compute” structures simply by binding to one another according to simple rules. In both cases there is no formal algorithm, but initial structure (input) is acted upon by rules to produce final structure (output). Is this much different than formation of structure by a cellular automaton computation that creates triangles of all sizes as described in chapter 2 and shown in figure 2.4? In all these cases, distinct programs do not exist, and what happens is dictated by rules acting on structures.

### How does a solar system form?

An example of equilibrium structure with motion is a solar system. Our solar system can also, for the most part, be understood as a consequence of three interacting laws. In this case they are the conservation of energy, the conservation of momentum, and gravity. The conservation of energy law states that energy can be neither created nor destroyed, but it may change form. Some familiar examples of this are the conversion of chemical energy to heat in a fire, the conversion of potential energy to kinetic [movement] energy when a sky diver jumps out of an airplane, and the conversion of mechanical energy to electrical energy that occurs in an electric power generator. The conservation of momentum states that an object in motion will continue with the same speed and direction unless acted upon by an outside force. In its classical formulation, the law of gravity states that two objects will attract one another with a force proportional to the product of their masses and inversely proportional to the square of the distance between them. The law of gravity can be expressed by a simple equation similar in form to Coulomb’s law except that masses attract instead of electrical charges, and masses never repel one another.<sup>6</sup>

The formation of a solar system like the one we live in is easy to understand if you don’t get bogged down in the details. The story goes like this. Stars have been a feature of our universe for a long time but none are as old as the universe and new ones are forming today. Before our sun existed, there was a cloud of gas and dust in a region of space. The cloud was mostly hydrogen and helium but contained some carbon, oxygen, nitrogen, iron, and all the other elements found in our

solar system today. The hydrogen and helium were mostly formed in the first few minutes of the universe after the big bang, while the other elements formed in stars whose own existence ended in supernova explosions scattering their constituent elements into space to become dust particles. These supernovae were important because they created the heavier elements and also because the outflow of material from each explosion caused parts of nearby dust and gas clouds to be in motion and to become unevenly distributed.

A lumpy gas cloud in space is not a stable object. The law of gravity dictates that every atom, molecule, and dust particle attracts every other atom, molecule, and dust particle. As a result, atoms tend to move toward other atoms, and clumps move toward other clumps. Countering this movement of everything toward everything is momentum.

For particles in motion, but not initially moving toward some center of mass, the conservation of momentum prevents them from proceeding directly toward one another. The interplay between the conservation of momentum and the law of gravity results in particles following curved paths spiraling around centers of higher density.

Energy must also be conserved. As the particles spiral toward one another, they speed up because gravitational potential energy is converted to kinetic (movement) energy. Over time, as the self-attracting gas and dust particles spiral closer and closer together, they move faster and faster and collide more and more frequently. What happens in the collisions depends on the chemical properties of the particles. Some collisions are elastic, where like billiard balls the particles bounce apart. Other collisions are inelastic, where the particles stick together like bits of putty. For all collisions the conservation of energy and the conservation of momentum must be obeyed.

Collisions also produce heat that is radiated off into space in the form of infrared radiation. The result of all the particles colliding and bouncing and sticking together is that sticky particles grow and all particles acquire coordinated motions. The lost heat energy results in everything moving closer together and slowing down. Because of rotation the gas cloud assumes the shape of a disk. Most of the dust and gas will be near the center with decreasing amounts further out. The conservation of energy and momentum dictate that the particles and gases near the center move the fastest while the material farthest away moves the slowest. Over time, as the molecules approach ever closer to the

center of rotation, they collide more and more frequently. More and more collisions mean more heat is released, and the center gets hotter and denser. Eventually, the center of the cloud gets so hot and so dense that thermonuclear reactions begin. This produces even more heat and other forms of radiation. The radiation pushes outward on atoms and molecules counteracting the inward pull of gravity.

When a balance is achieved between the outward push caused by radiation and the inward pull of gravity, a stable structure forms; we call it a star or sun. At about this time, the outer parts of the gas and dust disk become gravitationally unstable because the pull of gravity by the larger orbiting chunks changes the paths of nearby bodies. The altered orbits intersect, and the resulting collisions form ever-larger bodies that have larger and larger gravitational pulls on nearby aggregated bodies, gas, and dust. In relatively short order, the growing bodies effectively gobble each other up until only a small number of the largest aggregates (planets) remain. Planets each orbiting a sun in the grand orderly configuration we recognize as a solar system are neatly spaced simply because the great distances do not allow them to get at each other. If they were closer together they would pull each other out of orbit and eventually collide.

So that's it. The actual details are extremely messy, but the interplay of three simple rules virtually guarantees that ordered structure in the general form of a solar system will result whenever a gas and dust cloud exists that has suitable size, motion, and unevenness. Solar system details depend critically on the size and composition of the initial cloud, on "stickiness" rules (quantum mechanics), and on the rules that govern thermonuclear reactions. However, a basic understanding of the orderly structure we call a solar system is provided by considering the interplay of gravity, the conservation of momentum, and the conservation of energy—three simple rules. Because of the abundance of gas and dust clouds in our universe, we know solar systems must be common even though they have only recently been detected. Motion plays a central role in all solar systems because one of the rules, conservation of momentum, is only dispersive when objects are moving. Momentum opposes gravity when masses move in any direction not directly toward or away from one another, and planetary orbits represent a balance between a tendency to move in straight lines (momentum) and a tendency to move toward other masses (gravity).

Our solar system illustrates two types of structure. The stable orbiting of planets is equilibrium structure. The continued motion and spacing between planets requires no input of energy, and in the absence of friction or some outside force, the solar system will continue wheeling in space forever. The structure of the sun at its center is different. If the nuclear furnace were to go out, our sun would shrink to about one one-hundredth of its current diameter. If it were significantly bigger it would become a black hole. A star requires the continual generation of energy from thermonuclear reactions to maintain its stable structure. It is a nonequilibrium structure.

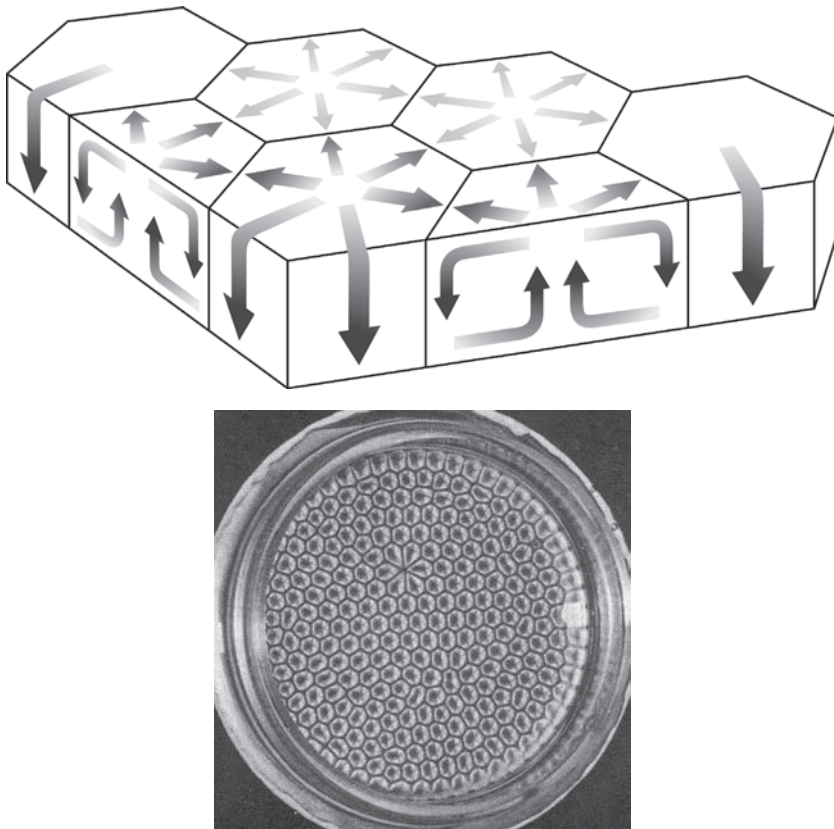
### How do nonequilibrium structures form?

Differentially heated fluids and gases provide a rather different example of spontaneously forming structure. Such structures are only slightly more difficult to understand than those resulting from a simple balance between attracting and repelling forces, but their formation seems more magical.

Anyone who can boil water can observe nonequilibrium structure. When a pan of water is heated from below, a remarkable transformation occurs. If the amount of heat is modest, there is nothing much to notice. At the submicroscopic level, water molecules colliding with the warm pan bottom rebound faster, acquiring kinetic energy in collisions with vigorously vibrating pan atoms. Energy is transmitted collision by collision up through the water column. So long as the heating is modest, the top of the water will be cooler than the bottom, but not by much. When the burner is turned up, more heat will be imparted to the bottom of the pan and the violence and rapidity of collisions will increase. Also the temperature difference between the top and the bottom of the water column will become larger.

Hot water is less dense than cold water, so when sufficient heat is applied an unstable condition develops with cooler denser water overlaying hotter less dense water. Earth's gravity pulls more strongly on the same volume of cool dense water than on hot less dense water, and when the difference is enough, cool upper water and hot lower water change places by bulk flow. Because of incessant collisions in liquids, individual molecules are not free to act as independent agents. Instead,

coordinated movement occurs. Water plunges coherently downward from the top and rises upward from the bottom in different locations. Under certain conditions the mechanism that best transports heat from the bottom to the top of the water column is a coordinated motion we call convection cells. If the liquid is very shallow, a particularly dramatic example can be observed. In this case surface tension dominates over buoyancy, and if the pan is very flat and the heat delivery very even, a strikingly regular pattern of hexagonal convection cells called Bénard cells will form spontaneously (figure 3.1). In this structure



*Figure 3.1.* Diagram and photograph of Bénard cells. The diagram shows a cut away view of several hexagonal convection cells. Arrows show the direction of motion. The photograph is a top view of a thin layer of silicon oil containing aluminum powder to make the cells easily visible. Permission to reprint the photograph provided by E. L. Koschmeider.



water rises in the center of each cell and descends around the edges. If conditions are not so regular, the cells will be less regular. If the liquid is confined between two flat surfaces instead of open at the top, rolls (called Rayleigh-Bénard rolls) will form rather than hexagonal cells.

These regular movements are quite stable and will persist as long as the right amount of heat (not too much and not too little) is applied to the bottom of the pan. With too much heat, the pattern will break up into chaotic motion. With too little heat, all coordinated motion ceases.

Convection cells are a classic example of nonequilibrium structure in which vast numbers of molecules spontaneously move in a coordinated fashion without any specific instructions telling them where to go. It is this quality that has led some scientists to propose that far-from-equilibrium self-organization will ultimately provide an underlying explanation for many of the complexities observed in living systems. This remains to be seen. Bénard cells are not very complex. Like crystals, they result from the interplay of simple rules. For thin liquid layers with air against the upper surface (conditions required to observe Bénard cells), the interacting phenomena are viscosity, surface tension (both aspects of attraction between molecules), and thermal motion (diffusion).<sup>7</sup> For thick liquids, buoyancy (based on density and gravity) dominates over surface tension and drives the convection.

A more complicated example of convection-based structure is a thunderstorm. Storms are similar in many ways to Bénard cells, but they occur in air instead of liquid, buoyancy is the driving force, and the environment is much less uniform than a flat pan on a stove. Atmospheric convection has additional features, air cools and expands as it rises, and water vapor changes from gas to liquid when the temperature drops below the condensation point. Also, heat is released when water condenses. A rising body of moist air can generate enough heat from condensation to cause it to rise further, and paradoxically, to cool further.

A tropical thunderstorm begins on a sunny morning in a location with a lot of water vapor in the lower atmosphere. Heat and light from the sun warm the ground or sea surface, which in turn, warms the lower atmosphere. As the day progresses, the lower atmosphere gets warmer relative to air higher up. Eventually, the atmosphere becomes unstable, and the warm air near the surface rises and the cool air above

sinks much as in a Bénard cell, but the presence of water vapor changes things. Rising air expands and cools according to the universal gas law. When enough cooling has occurred, water vapor condenses into droplets of liquid water, releasing heat. This heat warms the air, causing it to rise further and cool more as it expands; this in turn causes more condensation and more heat to be released. The rising air is replaced by a continual inflow of warm moist air from the nearby lower atmosphere, generating still more droplets. Small droplets collide and coalesce into larger droplets.

Eventually, the upward rising air cannot support the weight of the growing water droplets, and they begin to fall. As the drops fall, they entrain surrounding air molecules resulting in a rush of downward flowing air and raindrops. Water droplets pick up and carry free electrons, and droplet-loaded wind currents generate large charge separations between different parts of the cloud, amplifying natural differences between the cloud and the ground. When these charge separations become big enough, they are neutralized by concentrated current flows we observe as lightning. These current flows locally and violently heat the air, resulting in the production of light and sound. The rising and falling columns of water-laden air also generate horizontal and even rotating winds. The storm continues by drawing in warm moist air and dispensing cold water as rain (or snow if it is cold enough). When the supply of rising moist air is cut off, the storm dies. This most frequently happens when the sun goes down but may also happen when dry air is drawn in or the storm encounters warm air aloft.

Like crystals, Bénard cells and thunderstorms result from a few simple rules acting on initial conditions (input) that are not particularly unusual.

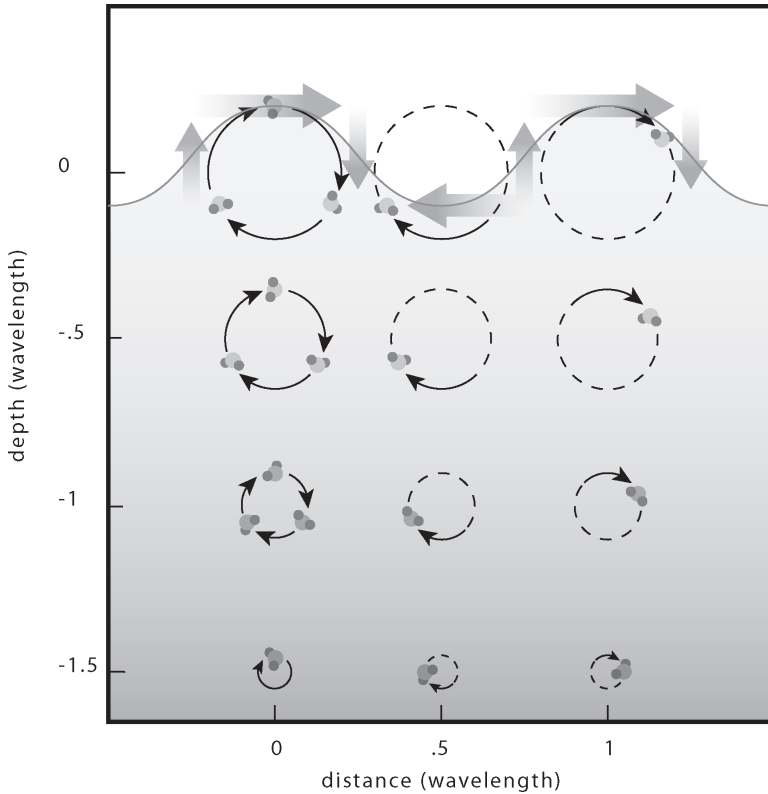
### Must structures have permanent parts?

Wind blowing across a lake or ocean forms waves, and the stronger and longer the wind blows, the larger the waves get. Water waves are nonequilibrium structures, and their formation is similar in some ways to the formation of Bénard cells, but the source of energy is wind. Similar to thunderstorms, the molecules that make up waves participate only temporarily. They are briefly part of a longer-lasting structure

that moves on leaving the individual molecules behind. In this manner waves and thunderstorms have a degree of independence from their parts (molecules). It turns out that many structures are like this including our own bodies. How can this be?

Water waves provide a good example. When wind blows across a flat surface, air molecules collide with surface molecules causing them to move. Molecules in a liquid cannot move far before colliding with other molecules. Because of microscopic irregularities, wind molecules impact some water molecules more than others. This causes them to “pile up.” As with Bénard cells, this movement is a coordinated activity of vast numbers of molecules. The initial pilings-up are called ripples, and once ripples form, molecules on the windward side inevitably experience more wind force than those on the leeward side. Because molecules are constantly bumping into one another, energy is transferred molecule to molecule across the surface. Because water is not easily compressed, every motion must be countered by a compensating motion. The way these various constraints are satisfied is for coordinated movements to spontaneously arise in which individual molecules trace out vertical circles. The summation of vast numbers of vertical circular movements is a wave. Molecules move around—forward, down, back, and up—as the wave passes (figure 3.2). Those at the crest move in the direction of the wind, and those in a trough move back. Each molecule rhythmically moves ahead, down, back, and up, never traveling very far even though the wave may travel a great distance. This coordinated motion satisfies the imperatives to move in response to wind force and to “fill in” when water at the top of the crest moves forward. Surface molecules never leave the surface; rather, the surface rises and falls as the direction of their motion follows circular paths. Molecules beneath the surface also move in circles, but in smaller and smaller circles as one looks further and further below the surface.

This highly coordinated motion of vast numbers of water molecules constitutes dynamic nonequilibrium structure. The longer and harder the wind blows, the larger the circles and the larger the wave structure gets. When the wind stops, the organized structure slowly dissipates and waves get smaller and smaller. Calculating the motion is complicated, but the rules that determine wave formation and propagation are quite simple, just cohesion (water sticking together), incompressibility, and transfer (conservation) of energy from wind to water and



*Figure 3.2.* Diagram of the coordinated motion of molecules in water waves. Arrows show the direction of motion in different parts of the waves. The waves are moving from left to right while molecule movements trace out circles.

water molecule to water molecule. Molecules participating in the wave structure are transient; they participate in one cycle and are left behind. It is as if the motion itself is an object. Once again simple rules acting on unexceptional initial conditions results naturally in the formation of structure.

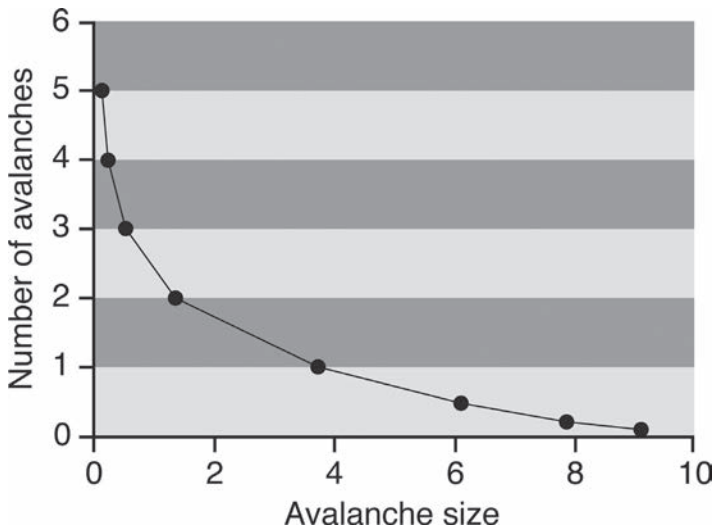
### Is behavior a form of structure?

Physical objects result from rules operating on matter in three-dimensional space; behaviors result from rules operating on structures over time.

Water waves change rhythmically, a simple pattern. More complicated behaviors are produced by systems that store up potential energy and release it intermittently. Some have irregular but characteristic behaviors. Lightning is an example, and so is an unstable snow mass on a steep mountain slope. When an unwary snowmobiler vibrates the snow bank, it may cascade down the mountainside in a very complex set of motions. Once the motion stops, however, the potential for future motion is gone until snow on the slope builds again to a critical state.

Physicists have studied the behaviors of avalanche-generating systems in the laboratory using sand instead of snow. It turns out that a sand pile exhibits surprisingly complicated behavior even if one does nothing more than slowly add grains to it. As more and more sand accumulates, the pile gets taller and the sides steeper. Eventually, one more sand grain causes an avalanche. Adding more grains will rebuild the pile until another avalanche occurs. The interesting thing about this system is that the sizes of the avalanches are extremely variable. Sometimes avalanches are large, but more frequently they are small. If one carefully measures the sizes of many avalanches, they obey what is known as a “power law.”<sup>8</sup> Larger avalanches are much less likely to occur than smaller ones. When power laws hold, the number of events of a particular size is less by a constant amount for each doubling of event size. An example is shown in figure 3.3. The power law relationship shows that the sizes of seemingly random avalanches are actually correlated in an orderly way. This correlation may be seen as a kind of structure.

Many phenomena exhibit behaviors that are characterized by power laws. These include such seemingly unrelated things as the size and frequency of earthquakes, the size and number of inlets on the coast of Norway, the ranking and frequency of word use in the English language, the size and frequency of biological extinctions, and the size and frequency of towns and cities. A more complete discussion can be found in *How Nature Works* by Per Bak.<sup>9</sup> Bak observes that power law relationships are characteristic of systems having a property he calls self-organized criticality (SOC). The idea is that situations are encountered where configurations are naturally poised between orderliness (crystal-like) and randomness (gas-like). In these special boundary regions, the laws of chemistry and physics lead to especially complicated states in which every element of the system influences every other element. These special states seem to occur more frequently than one



*Figure 3.3.* The nonlinear relationship between avalanche number and avalanche size, with small avalanches much more common than large ones. If replotted as the logarithm of avalanche number versus avalanche size, the resulting graph would fit a straight line sloping down to the right.

might expect because they “self-organize.” This simply means that some systems naturally progress over time until a critical state is reached. The sand pile provides an easy-to-visualize example; the critical state is poised at a balance point where one more grain of sand might cause an avalanche. The system cannot progress beyond the critical state because an avalanche makes it subcritical; adding more sand brings it back to criticality. Systems with this property tend to exhibit behaviors characterized by power laws and thus behave in a predictable (i.e., structured) way. Per Bak and others argue that SOC plays an important role in many complex behaviors.

No one denies that SOC is a phenomenon that characterizes some systems in the nonequilibrium domain, but it remains unclear how central SOC is to the overall understanding of complexity. As you drive your car, your odometer increases from zero to one to two to three, and so forth, until ten is reached, at which point the first column reverts to zero and a one is added to the second column. This is a kind of avalanche, where the first nine miles recorded in the “ones column”

are discarded and replaced by a one in the “tens column.” The pattern is repeated over and over until 100 miles is reached, at which point a larger avalanche occurs in which the contents of the first two columns are discarded and replaced by a one in the third column. If one plots the size of odometer rollovers against the logarithm of the number of times a rollover of each size occurs, you get a straight line indicative of power law behavior. An odometer exhibits a kind of SOC and illustrates powers of ten very well, but does not provide much insight into the subject of complexity.

The appeal of the sand pile example is that this simple system can generate significant behavioral complexity manifest by different-sized avalanches. Avalanches occur in a statistically predictable manner, but individual avalanches are unpredictable unless one keeps track of every grain in the entire sand pile.

### Can avalanches be simulated on a computer?

In chapter 2 we saw that there were no clear distinctions between some computations and some physical processes. Keeping with this theme, the concept of a sandpile can easily be reproduced in a computer. As with physical sandpiles, algorithmic sandpiles produce complicated outcomes that can be characterized by statistical power laws.

An example is shown in figure 3.4. The scheme works as follows. A computer program randomly adds (virtual) grains of sand to a “board” consisting of twenty-five (virtual) boxes. In the figure each large square is a board, and each small square with a number in it is a box. Each box can hold up to three grains. When the number of grains reaches four, these are distributed, one to each of the four neighboring boxes, leaving the original box empty (0 in the figure). When a box with four grains is located on the edge of the board, a grain distributed to the edge “falls off” the board and is lost to the system.

Starting with an empty board (all 0s), grains gradually accumulate until some boxes have three grains as shown in the upper right-hand board in the figure. When a new grain is added to a box with three grains, a dynamic sequence of events unfolds, the details of which depend on the exact configuration of all boxes on the board. In figure 3.4, board 2, a fourth grain is added to the center box. Distribution

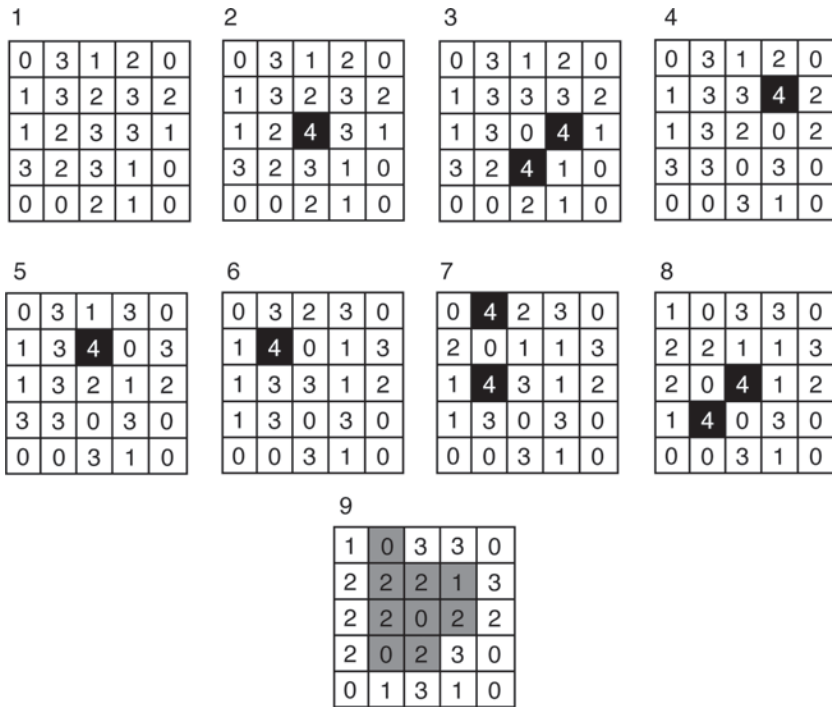


Figure 3.4. Avalanche model. In panel 2 a fourth grain has been added to the center box. The four grains are then distributed to the four neighboring boxes. This results in two new boxes with four grains (panel 3). These grains are then distributed to their neighbors. The avalanche continues until no boxes have four grains. The final configuration and all the boxes affected by this avalanche (shaded) are shown in panel 9.

of the four grains causes two neighboring boxes to have four grains (board 3), and distribution of these causes another box to have four grains (board 4). The process continues until there are no boxes with four grains (board 9). Such a sequence is called an avalanche, and this particular avalanche is characterized by ten overflow events from nine boxes. During the sequence the center box fills up and distributes grains a second time before the avalanche ends. Board 9 has no boxes with four grains and highlights all the boxes that experienced at least one overflow event.

Adding a grain has a different result depending on which box it is added to and the configurations of all the boxes making up the board.



The result may be no avalanche, if the grain is added to a box with less than three grains; an avalanche of size one, if the grain falls into a box with three grains but none of its immediate neighbors have three grains; or to larger avalanches, if one or more neighboring boxes have three grains. If one plays this game over and over with randomly generated initial boards and plots the logarithm of avalanche size against the number of avalanches, the result will be a straight line indicating a power law phenomenon. The upper limit of avalanche size is determined by the size of the board. Larger boards allow larger avalanches, but always the larger an avalanche, the less frequently it occurs.

This computer-based system illustrates again what is meant by self-organized criticality. No avalanches can occur until at least one box has three grains. A board with no three-grain boxes is subcritical. Once boxes with three grains are formed, the system is critical, avalanches can occur, but when they occur and what size avalanche depends on the exact state of the system and where the next grain falls. The system cannot become more than critical because four-grain boxes are not permitted. Thus, as long as new grains continue to rain down on the board, it is inevitable the system will “self-organize” into a critical state. The power law distribution of outcomes reflects the underlying probability of states that lead to avalanches of a particular size. In other words, states that allow large avalanches are much less common than those that lead to small avalanches.

The system demonstrates that power law type complexity can be generated by simple rules acting on specific initial configurations. Conceptually, the only difference between a computational sandpile and a real sandpile is that the “stickiness” rules are different. With real sandpiles, the laws of physics govern how many sand grains can pile up on one another before they fall down, while with sandpile algorithms the program specifies how many grains a box can hold.

## Whence structure for free?

The NKS (New Kind of Science) and tiling systems discussed in chapter 2 and salt crystals, solar systems, Bénard cells, thunderstorms, water waves, and avalanches discussed in this chapter are just a few of countless situations where structures and behaviors occur spontaneously

without direction in response to simple rules acting on particular initial conditions. Not all conditions lead to structure, but some do. The principle is the same whether the example is programmed in a computer, is an equilibrium physical structure, or is a nonequilibrium structure. These details do not make any great difference in our ability to understand, in a general way, how such structures come into existence. All of them illustrate a basic principle of structure formation: ***when appropriate simple rules are applied to certain situations, structure results as a consequence of interacting rules.*** This does not mean that any combination of simple rules and situations will cause structure to form—most do not—but rule combinations that result in some kind of structure are not that uncommon. They exist in nature and are not particularly difficult for people to invent in artificial systems.

In each of the six physical systems discussed in this chapter, if one were to use a super microscope to follow the behavior of a single atom or molecule during structure formation, the details would be wildly complicated, but when you step back and look at global behavior, the interplay of a few rules acting on particular but unexceptional initial conditions results in spontaneous formation of structure. Bénard cells, water waves, and thunderstorms all require a continual flow of energy through the system; crystals and solar systems do not. In the case of thunderstorms, irregularities in the surrounding atmosphere cause the structures to continually change. Sandpiles generate statistically predictable outcomes when grains of sand are added even though individual events are difficult to predict. Sandpile algorithms show that unpredictability is the outcome of vast numbers of interactions between individual sand grains. In critical states, all the parts affect all the other parts of a system. When this happens, behaviors become complicated.

Each example discussed in this chapter is characterized by simple rules and conditions that are not wildly improbable. What is remarkable, particularly for the nonequilibrium structures, is that molecules interacting with one another over very short distances ( $10^{-8}$  cm) form structures and behaviors having dimensions ranging from nanometers to billions of kilometers, depending on the system; and all this happens in the absence of any preplanning. There is no blueprint; simple initial conditions and the laws of chemistry and physics are all that are needed. It is truly structure for free.

I posed the following question at the beginning of this chapter: If physical processes compute, what and where are the programs? Reflecting on the examples, in every case physical laws operate on initial conditions resulting in new structures. Thus, initial conditions and the laws of physics and chemistry serve as inputs to the “computation.” There is no predesigned program, no instructions other than the rules in play, but these inputs are sufficient to result in significant structure.

In the next chapter, I turn to structures that depend no less on physical laws, but that differ because they require extra, often complicated local rules. These rules are so improbable they can never arise by random means. They require special explanation.

---

## Purposeful Structure

What can you get by using instructions that you can't get for free?

Imagine yourself to be an inventor and entrepreneur who comes up with an idea for a widget. You make a prototype out of string and carved pieces of wood, and it looks like it might work. How do you go about producing thousands of widgets made of aluminum and plastic that you can sell to eager widget buyers? You don't hire a theoretical computer scientist to design simple parts that when shaken together will spontaneously form completed widgets. Nor do you hire a nano-engineer to design microscopic robots that when programmed with simple rules can be set loose in a bin of aluminum and plastic blocks mindlessly reworking them into widgets. Rather, you carefully prepare blueprints and written instructions that can be followed by skilled machinists and mold makers and other instructions explaining how to assemble the finished parts so they will fit together in just the right way. Once you have appropriate instructions you can make lots of copies of this new structure.

In the last chapter we saw situations in which simple rules (the laws of chemistry and physics) applied to simple arrangements resulted in the spontaneous formation of structure. I call this “structure for free.” It is evident that lots of wonderful things from snowflakes to solar systems can form in this way. But, the structure-for-free concept (short programs applied repetitively to simple input) is not sufficient to explain blades of grass, Beethoven symphonies, and electronic computers.

There are a lot of things in our world that cannot be produced by employing simple rules acting on simple starting conditions. In computational terms, long programs are required. In the physical domain, long programs take the form of instructions, recipes, and blueprints that are focused on accomplishing a specific task. They encode purposeful information. Instructions are not limited to human activities, but are any kind of local rules that direct the formation of things. When such rules are followed, the outcomes are predetermined by the them. Such outcomes are characterized by a purpose, and the local rules (instructions) required to make them are never to be confused with the universal laws of physics. Thus, DNA (deoxyribonucleic acid) and RNA (ribonucleic acid) encode instructions that operate only in living cells, and blueprints only make sense to a machinist or a construction engineer. Instructions don’t substitute for the laws of chemistry and physics, they take advantage of them. And always, they have a purpose. Just how this special information makes a difference is best illustrated through example. Let us first explore the making of a common tool.

Few tools are as simple or as useful as a screwdriver, and although no handyman can do without one, few have actually made one. As common as they are, no one would suggest that a screwdriver is anything but the result of human activity. We can be quite certain that no screwdriver was ever the outcome of the laws of nature acting spontaneously on naturally occurring assemblages of matter. Why? To answer this question let’s look in detail at the creation of a screwdriver.

Nate Johnson, then a sophomore engineering student, actually made the tool I will describe in a beginning mechanical engineering laboratory course at Iowa State University. I define a screwdriver to consist of two parts: a hard tough rod with one end flattened so that it fits a slot in the head of a screw and a larger diameter handle that provides a good fit to the human hand. The larger diameter handle allows leverage to be applied, so a modest twisting force applied to the handle delivers

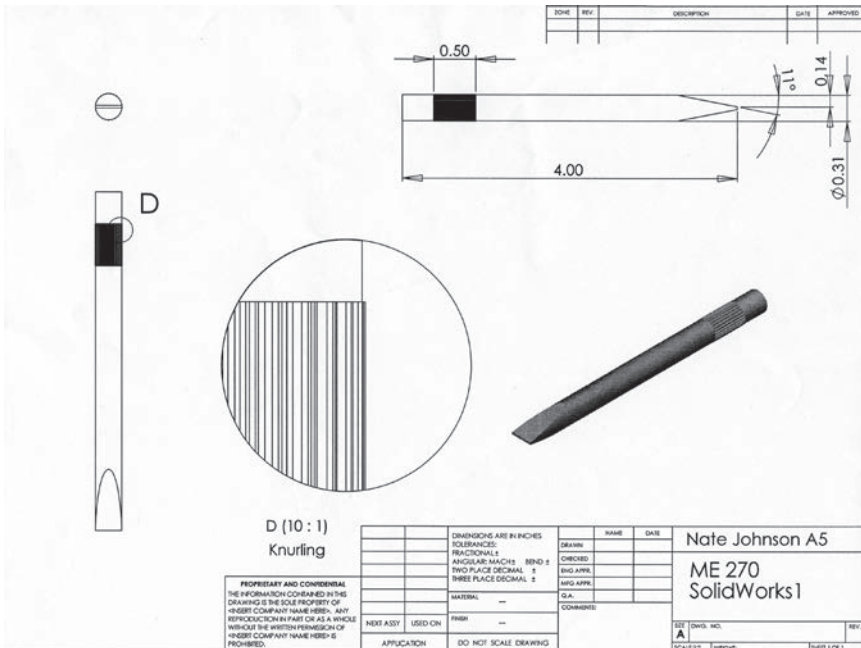
substantially greater torque to the flattened end. The fact that a variety of shapes are capable of turning screws does not detract from the essential nature of screwdrivers. Figure 4.1 shows blueprints and computer-generated drawings produced by Nate as part of his class assignment.<sup>1</sup>

Nate and his fellow students were provided with a 0.31-inch diameter steel rod and a 1-inch diameter plastic rod. Other materials could have been used. The shaft and blade must be strong but not brittle. Steel works best, though other metals such as bronze or titanium would work. The handle could have been made of wood or ceramic. In class students learned to use sophisticated power tools, but they could instead have used a hand drill, handsaw, file, knife (for shaping the handle), and even a rock for a hammer.

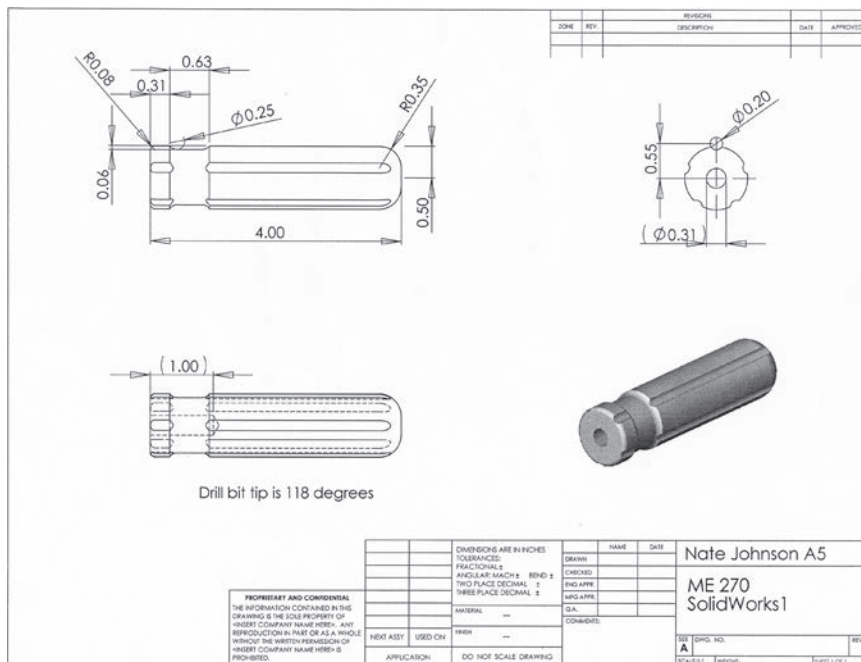
Blueprints provide visual guidance to the builder, and to use them Nate needed to create a series of construction steps. Box 4.1 lists such steps, instructions that, when followed, yield the screwdriver sketched in figure 4.1.

To make his screwdriver, Nate relied on the laws of chemistry and physics no less than nature does when a salt crystal forms. Both processes involve the formation of structure starting from initial configurations, but unlike salt crystals, configurations that lead to screwdrivers don't occur spontaneously, they are too improbable. The role of instructions is to specify a sequence of low probability configurations. Instructions are organized in steps where each step sets up a configuration that leads to an outcome necessary for the next step. In the special case of blueprints, the interpreter must translate pictures into steps. The builder, in this case Nate, is an integral part of these configurations. Once established, a configuration leads to a natural progression of events determined entirely by the laws of chemistry and physics (structure for free).

Let's examine the first step in detail. The steel rod, itself a very special configuration, is pressed against a moving saw blade at a spot four inches from one end. The distance from the end is irrelevant to physics, but physical laws determine everything that happens when a rod is steadily pressed against rapidly moving saw teeth. Each tooth tears away at the rod surface. Individual metal-atom-to-metal-atom chemical bonds are ripped apart by mechanical forces, and shavings of metal peel off from the surface of the deepening cut. The exact details depend on the force applied, the hardness of the metal, and the hardness,



*The shaft and blade*



*The handle*

*Figure 4.1. Blueprint and computer-generated drawings specifying a screwdriver. Permission granted by Nathan Johnson.*

**Box 4.1.**

Instructions for making a screwdriver.

Use a power saw to cut the 0.31-inch steel rod to a length of 4.0 inches.

Use a milling machine to shave (mill) one end of the rod at an angle of  $11^\circ$  and to a depth of 0.14 inches.

Rotate the rod along its long axis  $180^\circ$ , and shave the other side of the rod also at an angle of  $11^\circ$  and to a depth of 0.14 inches.

Using a knurling machine, knurl the rod  $\frac{3}{8}$  of an inch from the round end for a distance of 0.5 inches.

Cut the 1.0-inch diameter plastic rod to a length of 4.0 inches.

Use a lathe to make a thumb groove 0.08 inches deep and 0.63 inches wide, 0.31 inches from one end of the 4-inch plastic rod.

Drill a hole 1.0 inch deep and 0.31 inches in diameter into the center of the plastic rod at the same end as the thumb groove.

Use a stationary drill with a 0.20-inch bit to make five equally spaced length-wise grooves 0.05 inches deep for the length of the handle by moving the handle back and forth under the turning bit.

Use a hammer to tap the handle firmly onto the round end of the shaft so that the knurling embeds securely into the material of the handle.

sharpness, and speed of the saw teeth, but given the situation of a metal rod steadily pushed against sharp-moving teeth made of harder material, it is inevitable that a groove the width of the saw blade will be carved into and eventually through the rod. Once the initial configuration of a rod pressing steadily against the cutting edge of a moving saw blade is established, the action plays out according to chemistry and physics. This part of the process is structure for free.

Most of the steps in box 4.1 involve various cutting tools acting in much the same way but with different initial configurations. The last step is a little different. In this case, Nate positions the 0.31-inch hole previously drilled in the handle directly over the non-blade end of the steel rod and swings the hammer so it strikes the opposite end of the handle. Conservation of momentum and energy guarantees that force will be transmitted from the hammer to the handle, causing it to move down over the steel rod deforming the inner surface of the hole. If the force is great enough, where the rod is knurled, deformation will result



as plastic flows around and into the bumps and pits of the knurling. Once the motion stops, the handle is firmly fastened to the rod, and the screwdriver complete. The amount of force the hammer applies to the handle is critical. Too much force and the steel rod will be pushed completely through the handle, ruining it; too little force and the handle will not be firmly fixed onto the shaft. A hammer is a simple device that allows the builder to easily adjust the amount of force applied by repeated tapping. The important thing is that once the hole in the handle is placed over the end of the rod and the hammer set in motion, the forces of physics and material properties of the handle and steel rod entirely determine the outcome.

Each step specified in box 4.1 exhibits the same basic pattern. A particular configuration is established, and physical laws acting on that configuration result in a specific outcome predetermined by the details of the configuration and the laws of chemistry and physics. The nine instruction steps encode information needed by a person with prior knowledge of tools to create the necessary configurations in the correct sequence. The blueprint drawings contain additional information to help the builder interpret the written instructions. It would be possible, but less convenient, to express in English words all the relevant information contained in the drawings. Importantly, there are no simple rules or equations that preordain or describe the entire procedure. When a screwdriver is produced, the builder always follows an algorithm (a complicated rule) to achieve the desired result.

The nine-step rule given in box 4.1 is not the whole story. A complete accounting of the special information needed to make a screwdriver must include all the information needed to form the two rods, make the various tools and machines used, and directions for operating the machines. The complete instruction list must also encompass mining iron ore, making steel, manufacture and shaping plastic, and the manufacture of all tools and materials needed to make the tools used to make the screwdriver. An exhaustive compilation of the needed instructions would be very long indeed; but it is possible, at least in principle, to write down all instructions covering every aspect of the creation of a screwdriver.

When completed, even this listing would be incomplete. All of the instructions we have discussed so far are designed to be read, understood, and implemented by a human being. Aspects of a person that

are relevant to making screwdrivers are: a (human) body suitable for arranging pieces of matter and using tools and a brain for operating the body, understanding the instructions, interpreting blueprints, comprehending the task, and translating comprehension into action. Modern molecular genetics makes clear that the human body is itself largely the outcome of instructions. These instructions are written into the chemical structure of DNA molecules located in every cell in our bodies. Modern neuroscience makes clear that the internal information behind the reading of instructions, interpreting blueprints, understanding and using machines, and directing the actions of our bodies is encoded in neuronal connectivity and communication patterns in our brains. Written instructions, blueprints, DNA sequences, and neuronal connectivity patterns all encode purposeful information.

We now see a clear difference between a rock and a screwdriver. A screwdriver is special because of all the extra information required to specify particular sequences of improbable conditions needed for its formation. Only when an exacting sequence of events happens, do we get a screwdriver. Rocks, salt crystals, snowflakes, Bénard cells, and solar systems don't need extra information to form because the initial states they arise from are probable enough to occur spontaneously somewhere from time to time.

### Can you get a watch without a watchmaker?

Imagine an old-fashioned pocket watch, the kind with a windup spring and lots of gears inside. The coordinated actions of the various parts result in accurate time displayed on the dial. For a watch to work, just the right mechanical parts must fit together in just the right way, and for this to be so, each part must be made to exacting specifications and properly arranged. Living cells are more complicated than watches but they too utilize exactly shaped parts that fit together in just the right ways. Cell parts are not metal gears and springs, but proteins, large molecules essential to life. Because cells are more complicated than watches, they have more parts with more varied functions, but always each protein must have exactly the right properties for the job it does.

The watchmaker analogy has a long history. In 1802 theologian William Paley described walking across a heath and finding a watch.

Contrasting the watch with a nearby stone, which for all he knew could have been there forever, he asked how the watch happened to be in that place. Concluding it absurd that the watch, like the rock, might have been lying there since the beginning of time, he went on to argue that a watch must have a maker and, by the same logic, so must a human being. This famous imagery was picked up by Richard Dawkins in the title of his book *The Blind Watchmaker*.<sup>2</sup> In that book Dawkins argues that the watchmaker responsible for life is, in fact, the process of evolution—not at all what Paley had in mind!

To fully appreciate Dawkins's argument, we need to explore how cells work, and more specifically, how proteins are made. Before the modern era, all proteins were made by living cells. To understand life you must understand cells. These small organizational units have seven basic features, listed in table 4.1. Each depends on the activities of proteins.

Proteins function by binding to something. Like all very small things, they are chemical, and like all chemicals their surfaces have atomic scale features that attract or repel other atomic scale features. This means that when a protein fits precisely with another molecule and the corresponding surface features attract, the two structures will stick together. If the shapes don't match and attracting chemical features don't line up, no binding will occur. Precision binding requires 3-D positional specification at the level of individual atoms that is far more exacting than required for gears to mesh in a watch. How do cells make such structures?

Table 4.1

Seven features shared by all cells.

---

1. A lipid-protein semifluid membrane separating inside from out.
  2. DNA molecules encoding instructions for the synthesis of RNA and proteins.
  3. Machinery for the synthesis of proteins.
  4. A self-regulating metabolic network assuring a relatively consistent internal environment.
  5. Systems providing for the orderly replacement, repair, and duplication of all parts as needed.
  6. Mechanisms for cell growth and reproduction.
  7. Mechanisms for acquiring chemical energy and needed raw materials.
-

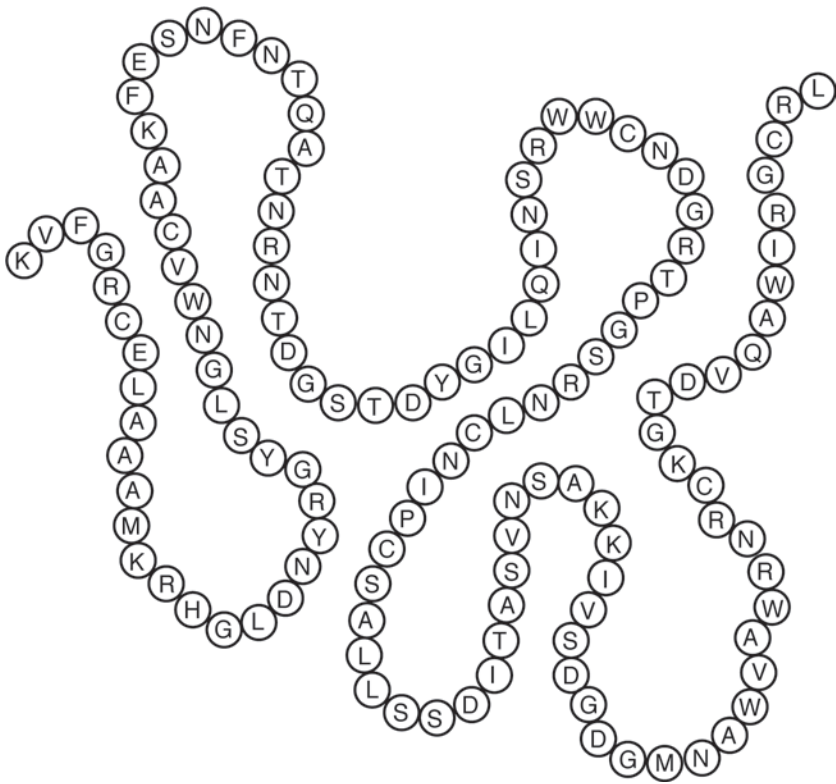
During the process of protein synthesis, information stored in DNA plays the same role in a cell that written instructions do for a human watchmaker. Instructions, written in the language of DNA, direct cellular machinery to assemble exactly specified protein molecules. The strategy is quite ingenious. Many steps involve self-assembly—structure for free, but the overall process cannot occur without instructions encoded in DNA.

Proteins are chains of smaller units known as amino acids, connected together by chemical bonds, much like beads that snap together to make a child's necklace. Amino acids are relatively simple chemicals ranging in size from ten to twenty-seven atoms (for comparison, water has three atoms and ethyl alcohol has nine). All cells in all kinds of creatures utilize the same twenty amino acids, and an average-sized protein has several hundred amino acids while a large protein has several thousand.

Amino acids share a common structural motif attached to a variable part, called the side chain. Chemical properties of the side chains are responsible for amino acid differences. Within a chain amino acids interact with (bind to) one another causing the chain to fold back on itself and the overall protein to take on a specific shape. That shape, and the chemical properties of whatever amino acids are on its surface, determine what a protein can bind to and how strongly. Figure 4.2 shows a chain of 129 amino acids linked together prior to folding, and figure 4.3 shows the same chain folded into its natural three-dimensional shape.

In nature, all proteins are synthesized in the same way. Information encoded in DNA is used to specify the order of amino acids in each protein chain; and in turn, the order determines the shape. The protein lysozyme, shown in figures 4.2 and 4.3, always has the same amino acid sequence, the same shape, and the same side chains exposed on its surface. If you were to change even one of the amino acids, the shape and surface details would be somehow different.

Some proteins catalyze (facilitate) chemical reactions between smaller molecules. Proteins with this property are called enzymes. Enzyme function requires binding to smaller chemicals that participate in the reaction. Other proteins function by binding to DNA, yet others by binding to specific proteins. All these binding functions are precise, and any change in the amino acid sequence of a protein may alter

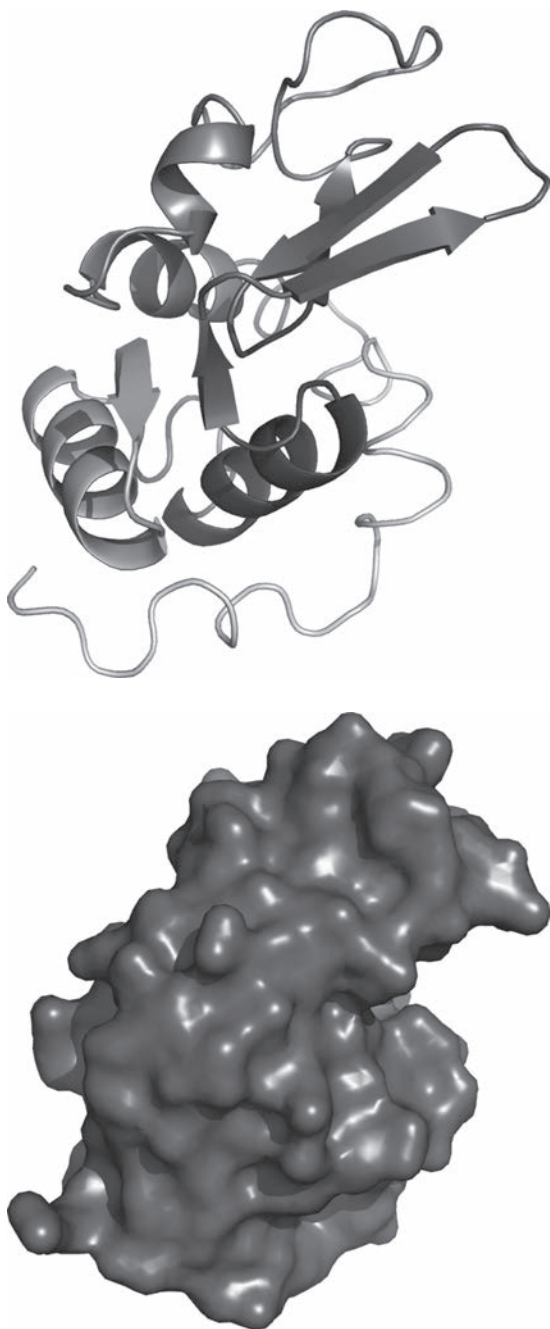


*Figure 4.2.* The amino acid chain of the enzyme lysozyme showing all 129 amino acids.

its detailed surface structure resulting in failure to bind to its original target and, possibly, it binding to something else. Human cells produce multiple copies of thousands of different proteins, and all bind to something inside or outside the cell with high specificity.

Because cells depend utterly on myriads of precise protein-binding events, synthesis of the proper proteins is essential to every living thing. This is achieved by miniature molecular machinery, which is itself made of protein and RNA, another class of large molecules related to DNA. The synthesis of RNA requires protein enzymes, and the synthesis of proteins requires RNA. Thus, the cell makes proteins using machinery that either depends on proteins or is made up of proteins.

The story of protein synthesis begins with DNA. Like proteins, DNA molecules are chains of smaller units. While proteins are made of amino



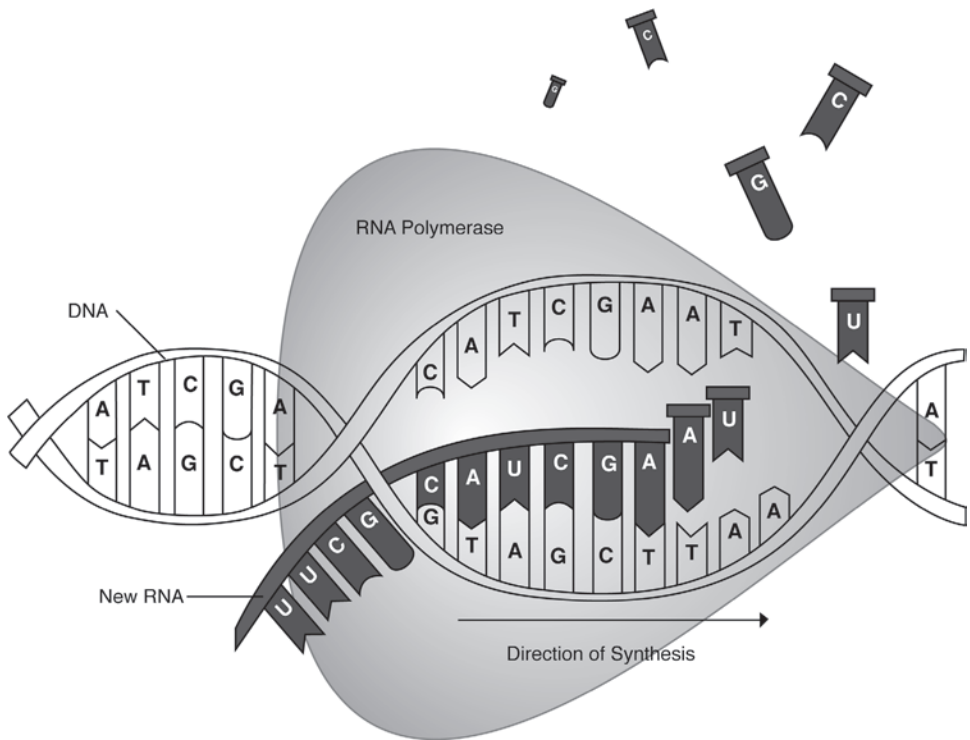
*Figure 4.3.* Two depictions of the three-dimensional folded structure of lysozyme. A ribbon model (top panel) reveals the path of the amino acid chain. A surface model (bottom panel) shows all the atoms on the surface of the molecule. Figures drawn with The PyMOL Molecular Graphics System, Version 1.2r3pre, Schrödinger, LLC.

acids, DNA molecules are made of nucleotides. There are four different ones. For simplicity it is customary to abbreviate their names as A, G, C, and T. Any order of nucleotides in a chain is permitted. Thus, a short DNA molecule might be shown as ACGATTCAAAGTCTCAG, where each letter stands for a nucleotide in the DNA chain. The letters spell out messages that are meaningful to the cell just as letters in English spell out messages that may be meaningful to you. DNA molecules in cells are extremely long, typically ranging from 1 million ( $1 \times 10^6$ ) to 100 million ( $1 \times 10^8$ ) nucleotides in length. Each is organized into functional units called genes, and each gene codes for a different protein.

DNA in cells consists of two nucleotide chains linked in a tight embrace called a double helix. This structure is made possible because A nucleotides bind to T and G nucleotides to C. This paring, in turn, means the sequence of each strand of a double helix is complementary to the other: whenever one strand has an A the other has a T, and whenever one has a G the other has a C. This redundancy allows damage to be repaired and for DNA molecules to be copied.

The first step in the synthesis of a protein begins with the synthesis of an RNA copy of a specific portion of one of the DNA strands. RNA is chemically similar to DNA but is made of nucleotides that are subtly different. They too are given single-letter nicknames, but in this case they are called A, G, C, and U. A protein enzyme, called RNA polymerase, creates a new chain of RNA nucleotides that has exactly the same sequence as one strand of a section of the larger DNA except that where DNA has T, the RNA copy has a U (figure 4.4). Most of these copies are called messenger RNA (mRNA), but there are also other classes of RNA that have other functions.

Messenger RNA molecules bind to molecular-scale machines called ribosomes. The job of a ribosome is to synthesize proteins by successively linking together amino acids in an order specified by the order of nucleotides in mRNA. In this way the order of nucleotides in DNA indirectly, but exactly, determines the order of amino acids in each protein. Because there are four RNA nucleotides and twenty amino acids, the specification rule cannot be one amino acid per nucleotide. The cell uses a triplet code where three consecutive nucleotides in an mRNA chain specify one amino acid in a protein chain. There are sixty-four different ways that four nucleotides can be arranged in groups of three (AAU, ACU, GAU, etc.); each unique triplet is called a codon.



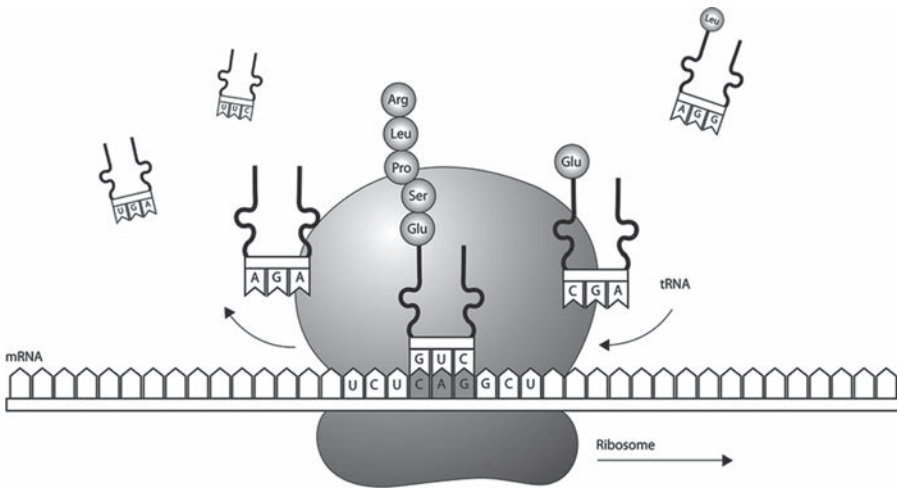
*Figure 4.4.* Artist's portrayal of RNA polymerase making an RNA copy of a DNA sequence.

Sixty-one of the sixty-four possible codons specify a particular amino acid. The remaining three are “stop” codons. The process is called translation because information encoded in the language of nucleotides is “translated” to information in the language of amino acids.

Because amino acids do not bind efficiently to nucleotides, adapter molecules called tRNA serve to align the proper amino acid with its corresponding mRNA codon inside the ribosome (figure 4.5). These adapters ensure that the amino acids line up with nucleotides read three at a time. It is important that this process work correctly. If an error is made, the resultant protein will have an incorrect amino acid in its sequence, and it may fold into the wrong shape and fail to function properly.<sup>3</sup>

As the growing amino acid chain is synthesized, it protrudes from the ribosome and amino acids interact with one another causing the





*Figure 4.5.* Synthesis of a protein showing the ribosome, mRNA, and tRNA adaptors. The shaded structure is a ribosome interacting with mRNA (the long chain-like structure) and a tRNA that is base-paired to mRNA and also attached to a chain of amino acids. In the figure the next tRNA to bind is just entering the ribosome from the right, and the previous tRNA is exiting to the left. The chain of amino acids attached to the central tRNA will be transferred to the amino acid on the incoming tRNA after it base pairs to the next three nucleotides of the mRNA.

still-incomplete protein to fold up into a sequence-determined three-dimensional shape. For many proteins, when synthesis is complete, the protein has already assumed its final shape; others undergo subsequent modification, and still others achieve their final shape only when they bind to their target.

Spontaneous folding is key to understanding how information encoded in an amino acid chain is transformed into a three-dimensional structure. Each amino acid interacts with other amino acids, and the chain finds a shape that optimizes the total energy of all interactions. This part of the process is structure for free. A huge number of protein shapes are possible because there are a huge number of possible amino acid sequences and each folds into a distinct shape. This is how all of the precisely made proteins needed for proper cell function are made. Each is specified by instructions encoded in cellular DNA. It is

these DNA sequences that assure the right proteins are made. Change a sequence, and you change a protein. Living creatures are not watches, but are much more complex, and they come into existence all the time without the intervention of a watchmaker.

## How do instructions encoded in DNA specify structures more complicated than single proteins?

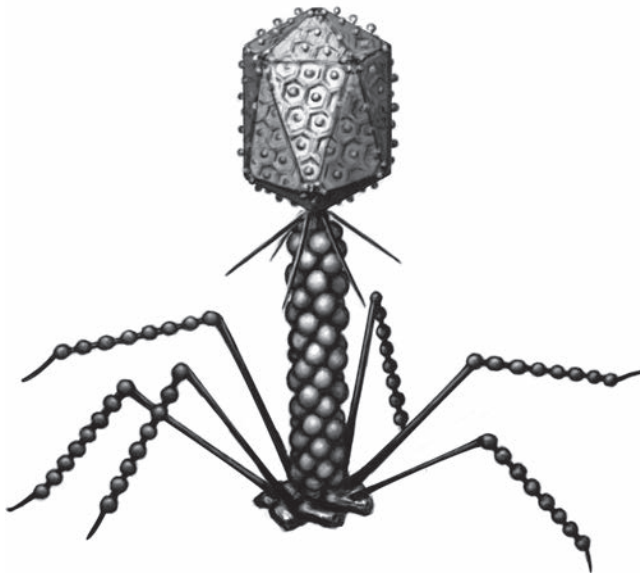
Let's continue the watch theme, but now imagine that the mechanical device was not assembled by a watchmaker but by placing the appropriate parts in a box and shaking them. Ridiculous? Could a watch be designed that would self-assemble if the parts were shaken together? What kinds of parts would be required? Surely not watch parts as we know them, but this is exactly how a virus forms.

A virus is a biological device fully as complicated as a watch, but much smaller. A virus is assembled by the molecular equivalent of shaking its parts in a box. How can this be? First of all, because the parts are very small, when suspended in water they spontaneously jump around by thermal motion, a phenomenon that characterizes all extremely small things. Second, the parts are proteins, and as we have seen, all proteins have very precise shapes. Third, their surfaces have molecular features that interact with other virus proteins. When the fit is right, each protein binds tightly to its matching protein.

Viruses reproduce, but do not grow and have no chemical metabolism independent of the cells they infect. During most of their existence they are completely inert, existing in the borderland between living and nonliving. Reproduction of viruses is very much like the assembly of a plastic toy where parts are snapped into matching slots in other parts. The difference is there is no mom or dad to assemble it. Like all proteins, each virus protein has a shape dictated by its particular sequence of amino acids. This shape (and associated surface properties) determines exactly which other shapes it binds to. All the information necessary to place a virus protein in its proper place in a partially formed virus is encoded in the details of its structure.

There are probably no living cells that cannot be infected by some virus somewhere. Fortunately, viruses are highly specific, and any particular virus can only infect a limited number of cell types. Viruses

that infect bacteria are particularly easy to study. They are called bacteriophages, or simply, phages. One of the most intensively studied bacterial viruses is a bacteriophage known as  $T_4$ . It consists entirely of protein and DNA. As viruses go, it is relatively large and complicated, all the better for illustrating the potential of self-assembly.  $T_4$  DNA encodes 274 genes that specify 274 different proteins needed for the assembly and reproductive cycle of the virus. Some of these proteins are enzymes that alter the metabolism of the bacterial cell it infects; some are enzymes required for the synthesis of  $T_4$  DNA; others are involved in the expression of the 274 genes. About fifty proteins have the job of building a remarkable outercoat structure that protects the DNA when it is released from a dying bacterial cell. The protein coat also orchestrates the infection of another bacterial cell when a suitable one is encountered. The  $T_4$  coat consists of over 3,500 protein molecules of forty different kinds. Figure 4.6 is a drawing of the structure. The DNA is hidden in this picture, coiled up inside the “head.” Luckily this particular virus cannot infect you or me or any human cell.



*Figure 4.6.* Artist's drawing of a  $T_4$  bacteriophage. (see note 4).

More remarkable than the overall structure is how it forms. There are no little elves scurrying around with tiny hammers putting all the parts together. The structure self-assembles. Each of the 3,500 protein molecules fits only with particular partner molecules. This, in itself, is not so remarkable; after all, the parts in your automobile also fit together in a specific way. But your car does not self-assemble. Car parts are not designed so that if you put them in a big box and shook them, they would come together as a car. Proteins are so small they naturally jiggle around in constant motion due to constant buffeting by even smaller, more rapidly moving water molecules. When matching shapes come into contact, they stick together. Every binding event creates a new binding site for another protein to bind. This establishes an assembly sequence, in which a third protein can only bind after the first two have bound together, and the fourth can only be added after the third, and so forth. In this way proteins add to the growing structure in specific sequences; and from a randomly mixed solution of proteins, specific structure spontaneously arises.

The assembly proceeds along three parallel “assembly lines” (figure 4.7), with head, tail, and long tail fibers assembling separately before finally coming together to form a mature bacteriophage. Before the head connects to the tail, it is filled with  $T_4$  DNA in a process that utilizes energy supplied by the infected cell.

The process is much like the self-assembling jigsaw puzzle illustrated in figure 2.10, but the structure is three dimensional and the pieces are proteins. Formation of both jigsaw and virus rely on purposeful information that specifies shapes that come together in precise and pre-determined ways. For the jigsaw puzzle the information specifying the design of pieces was created in a person’s brain. For  $T_4$ , the information is encoded in virus DNA. Without instructions to specify the details of each piece, viruses (or self-assembling jigsaw puzzles) would never come into existence.

Self-assembling viruses belong to a family of activities that include molecular tiling and NKS (New Kind of Science) systems discussed in chapter 2. Such systems compute physical patterns and structures by assembly. The “rules” of the computation, the program, are built into shapes and chemical binding properties of the assembling parts.

Much of the activity in cells is also based on the activities of self-assembling protein structures. In fact, an entire cell or even an entire

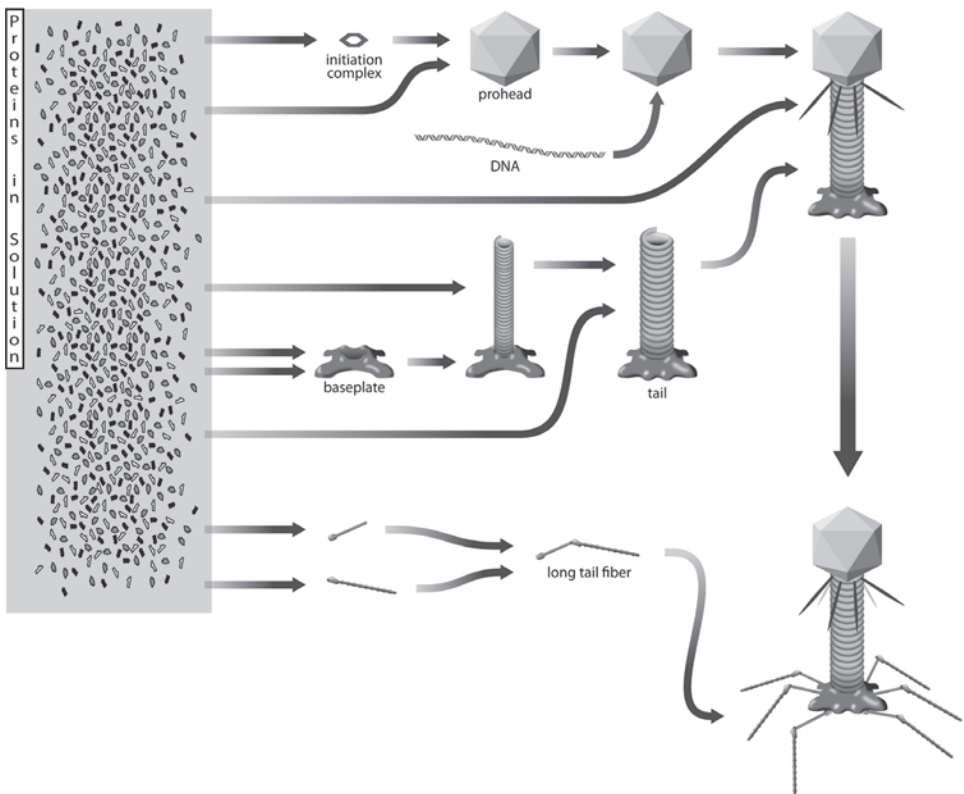
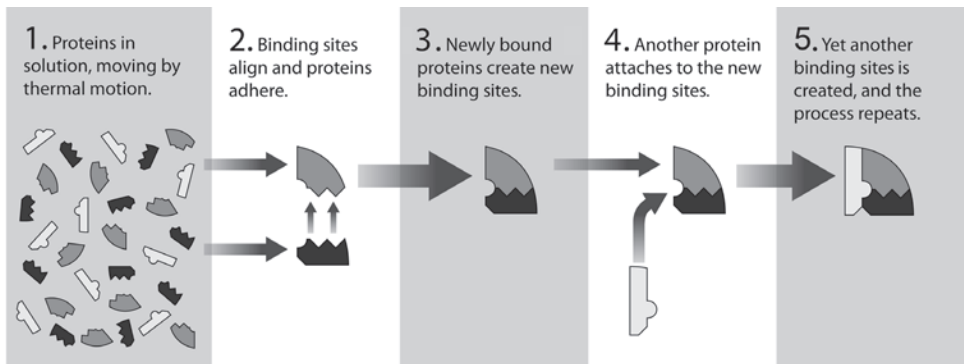


Figure 4.7. Assembly of T4 bacteriophage (see note 4).

body can be seen as the outcome of vast numbers of self-assembly events. Small parts come together in specific ways to create larger parts that, in turn, serve as the basis for still larger, more complex structures. Predetermined structure builds on preexisting structure. In this sense it is possible to speak of cells as living computers. Nearly all structures in every cell depend ultimately on instructions encoded in the language of DNA that prespecifies protein and RNA parts, how they interact, and what they form.

### Can instructions dictate activity?

Being small, viruses in water move about randomly, jiggling this way and that. Given time and luck, this random movement eventually brings a virus to a suitable host bacterium. At the tip of each of the six long tail fibers (figure 4.6) is a binding site that matches a specific molecular structure on the outer surface of a host cell. Once a long tail fiber protein binds to its target, in quick succession the other five also encounter nearby targets on the bacterial surface. In this configuration,  $T_4$  is firmly attached to the bacterium with its tail pointed down toward the cell and head up. Successful binding by the long tail fibers sets in motion a succession of protein conformational changes. The process is very reminiscent of the children's toys called "transformers," where parts are moved in specific sequences causing a car or spaceship to be transformed into a robot. The toy can be transformed back—a virus cannot.

In response to the long tail fibers binding, proteins known as short tail fibers located at the base of the tail unfold and bind to other features on the cell surface. This, in turn, causes the base plate to change from hexagon to star shape, triggering each of 144 tail sheath proteins to flatten. This shortens the tail sheath by almost two-thirds, driving the inner tail tube, syringe like, through the outer membrane of the cell. Suitable host bacterial cells have three outer layers, an outer membrane, a cell wall, and an inner membrane. If  $T_4$  DNA is to successfully enter the cell, it must leave the bacteriophage head, traverse the tail tube across all three cell barriers, and enter the bacterial cytoplasm. Contraction of the tail sheath forces the tip of the tail tube against layer two, the cell wall. This is a tough rigid structure. Enzymes at the tip of the tube break chemical bonds in the cell wall, weakening it and allowing

the tube to penetrate through. This places the tip of the tail tube in contact with the inner cell membrane. Tube proteins then interact with proteins already present in the inner membrane, forming a pore that facilitates passage of the DNA across the last barrier and into the cell cytoplasm where bacteriophage protein synthesis commences.<sup>4</sup>

During the binding and injection phase of its life cycle,  $T_4$  bacteriophage is briefly a very active structure with numerous proteins changing conformation and two large-scale rearrangements, sheath contraction, and DNA expulsion.  $T_4$  is not capable of generating metabolic chemical energy, so potential energy is stored in its structure at the time of assembly, much like the spring in a windup toy puppy that skitters across the floor when released. Once that stored-up energy is expended, the virus coat will never be active again. Unlike the toy, there is no spring to be rewound.  $T_4$ 's dynamic motions are entirely prescribed by its structure and by interactions of its proteins with each other and the environment. The detailed dynamic motions that play out during the infection process are prespecified by information encoded in  $T_4$  DNA. Specific external signals trigger predetermined motions. We can think of the motions themselves as being preassembled, waiting to happen.

### What about ongoing activity?

Bacteriophage infection involves a brief prescribed sequence of movements resulting in the virus structure achieving a lower energy equilibrium configuration. We next ask if long-lasting activity can also be preprogrammed. The answer, of course, is yes. Our example also comes from biology. It is a cell-based system that produces repeated waves of activity. These waves, called action potentials, consist of alternating patterns of ion flows and are how nerve cells in our bodies communicate with one another. The wave patterns constitute nonequilibrium structure occurring over time, and they require a continual input of energy from cellular metabolism. Ions, recall, are electrically charged atoms or molecules. This example also establishes a molecular basis for our later discussion of human learning.

Cells in the body are specialized to perform different tasks. Muscle cells specialize in the generation of physical force, red blood cells in the transport of oxygen, and white blood cells in the detection and

elimination of foreign cells and viruses. The job of nerve cells is long-distance communication within the body. The properties and activities of any cell are largely determined by the proteins present in them, and as we have seen, the proteins present are determined by which DNA sequences (genes) are expressed as mRNA molecules and then translated into amino acid chains that fold up into highly specific structures.

All cells maintain an internal environment different from that outside. Among these differences are concentrations of various ions. We are particularly interested here in sodium and potassium ions. In the body, sodium ions, the same as found in table salt, are much more prevalent outside of cells in the blood and interstitial fluids, while potassium ions are maintained at higher concentration inside cells. Specialized proteins called “pumps” located in cell membranes maintain these ionic differences by exchanging sodium ions inside for potassium ions outside. They do this by flipping back and forth between two structural configurations. Each configuration change carries ions across the membrane. One flip takes sodium ions out and the next brings potassium ions in. These pumping actions require energy, which is supplied by chemical reactions carried out in the cell by other protein enzymes.

Many nerve cells (neurons) are very long and thin. The longest in your body extend from your lower spine to the tips of your toes. In giraffes and giant squids they can be several meters long. Signals are transmitted along these cells by means of traveling action potentials. Each action potential consists of a brief inrush of sodium ions followed immediately by an outrush of potassium ions. This pattern moves down the cell. Associated with this transient pattern of ion flows is a voltage spike. In informational terms, an action potential can be thought of as transmitting one bit of information. Complex patterns of action potentials may encode complex messages.

At the molecular level, an action potential is not so complicated. It results from sequential openings and closings of small protein doors called “channels” in the cell membrane. These channels are structurally different from the sodium/potassium pump proteins, and they selectively allow ions to pass through without a requirement for external energy use. Two kinds of channels are relevant to our story, one specific for sodium ions and another for potassium ions. Nature always works to equalize chemical differences, so if both types of channels were open for a long time, sodium and potassium ion concentrations

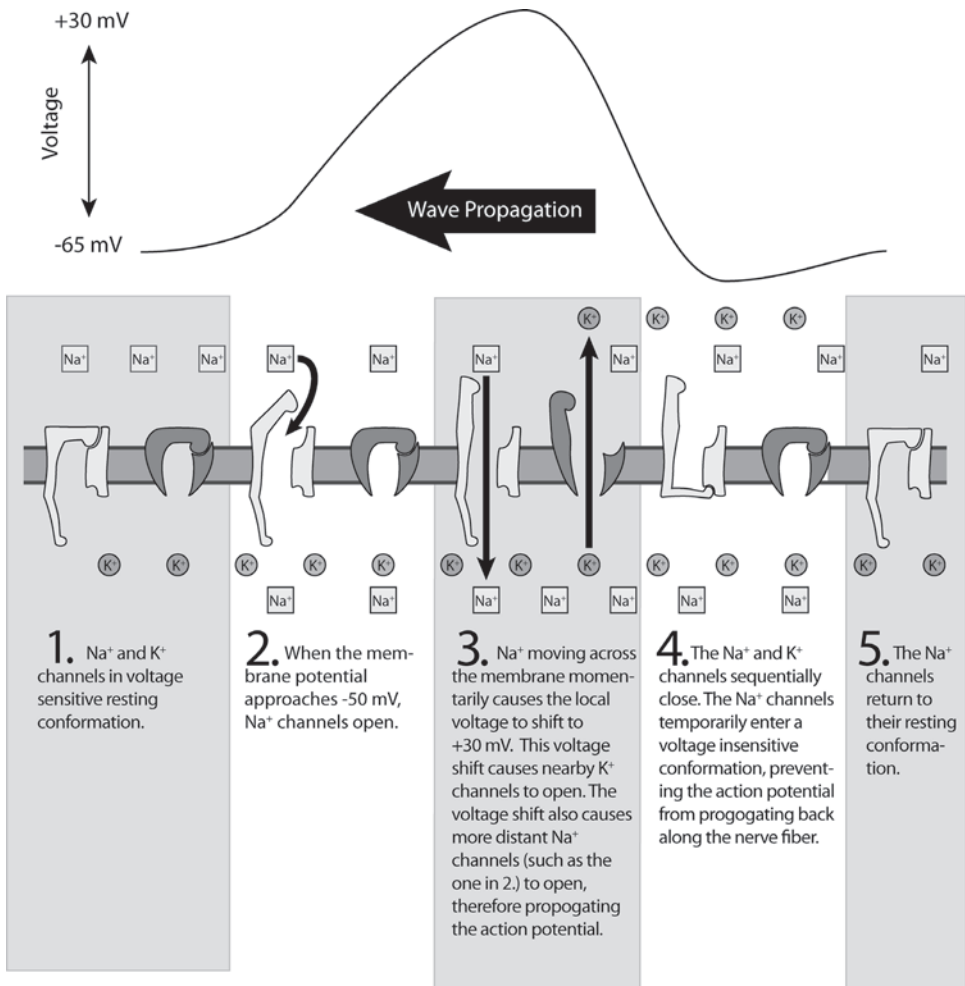


would equalize inside and out, and the cell would die. Since the door openings are brief, the cell is not harmed.

The combination of selective ion pumps and ion leaks creates a small electrical voltage across nerve cell membranes. This is called the “resting potential.” It is usually  $-60$  to  $-70$  millivolts (mV). A millivolt is one-thousandth of a volt. Ion channels in neurons open and close in response to voltage. They are said to be “voltage gated.” Action potentials are triggered whenever the resting potential of the cell becomes closer to zero than a threshold value, usually about  $-50$  mV. This type of voltage change is normally caused by interactions with other nerve cells, but scientists can also make them happen.

When the threshold is reached, voltage-gated sodium channels undergo a sequence of conformational changes passing through three successive structural states. A closed channel sensitive to voltage is called the resting state. When the membrane potential diminishes to  $-50$  mV, channels open, allowing sodium ions to enter the cell. This state lasts about one millisecond (one-thousandth of a second). The channels then spontaneously close, entering a temporary state that is insensitive to voltage. After about two milliseconds, the proteins reconfigure again to their original state. The entire sodium channel cycle takes three milliseconds. The short delay imposed by the second state is critical to the formation of action potentials.

The inrush of sodium ions locally alters the ion balance so that voltage across a small region of the membrane briefly changes from  $-50$  mV to  $+30$  mV. This change has two effects. It triggers voltage-gated potassium channels in the immediate vicinity to open, and it triggers sodium channels slightly farther away to begin to open. By the time the first opened sodium channels are closing (after being open for one millisecond), potassium channels in that region are opening. This allows potassium ions to flow out of the cell in that local region (carrying positive charge with them) returning the local membrane potential to normal. The timing is such that whenever potassium ion channels are fully open, the local sodium channels are closing, while slightly more distant sodium channels are beginning to open. Always, potassium ion channels open after sodium ion channels. The effect is an alternating inrush of sodium ions followed closely by outrush of potassium ions. This ion flow pattern migrates along the cell surface creating a temporary wave of voltage change. The process is diagrammed in figure 4.8.



*Figure 4.8.* Ion fluxes in the cell membrane responsible for voltage changes that characterize an action potential. The voltage wave is moving from right to left. (see note 5)

The entire cycle from sodium ion inrush, to potassium ion outrush, and return to normal takes about five milliseconds. Physically, action potentials move across cell surfaces. In figure 4.8 the voltage change occurring in a small region of a cell is shown. Sequentially, sodium ions rush into the cell as the voltage is rising (becoming less negative, or positive) followed by potassium ions rushing out as the voltage is falling.

Because the conformation cycle of each voltage-gated sodium channel takes several milliseconds to complete, a wave (sodium ion inrush followed by a potassium ion outrush) can only migrate away from the initial site of activation. Just after a wave has passed, sodium channels are temporarily insensitive to voltage. This period of voltage insensitivity guarantees that action potentials can only proceed one-way.

If nerve cells were spherical and large enough, waves would propagate around and around them, but neurons are long and thin, so propagation can only proceed along a cell until an end is reached.<sup>5</sup> Neurons branch and action potentials proceed out to each branch tip without being diluted. At every terminus are special structures called synapses that contact other neurons, allowing action potentials in one cell to modify the resting potential of the neurons contacted. The effect of synapses on other neurons can be either excitatory or inhibitory. Every neuron integrates all the signals arriving to it at about the same time and initiates a new action potential whenever its resting voltage reaches the threshold. In this way action potentials are selectively transmitted from neuron to neuron. When negative inputs to a cell outweigh the positive, incoming signals are not passed on. This allows the system to process information.

Action potentials constitute a dynamic renewable pattern that results from the coordinated activities of proteins having very specific properties located in nerve cell membranes. Key proteins are sodium/potassium ion pumps maintaining the resting potential and voltage-gated sodium and potassium ion channels whose cyclic conformational changes generate sequential ion flows when triggered. The detailed properties of these proteins, especially the cyclic conformational changes experienced by the voltage-gated sodium channels and the characteristic times these channels remain open before closing, make action potentials inevitable whenever proteins with these particular properties are present in a cell membrane and the membrane potential rises to the threshold value. Because action potentials result from the activities of specific proteins whose structures are specified by DNA sequences, they can be considered predetermined by instructions. As with bacteriophage infection, dynamic activity results ultimately from instructions for building specific protein structures that are encoded in DNA sequences.

## So, exactly how do we define purposeful structure?

Examples given earlier, in chapters 2 and 3, show convincingly that significant complexity is possible without instructions in both static and dynamic systems. This chapter provides examples of various complexities made possible only through the use of instructions. What is the difference? By most measures a thunderstorm is more complicated than an action potential and certainly more complicated than a screwdriver. Yet, in some ways a thunderstorm is simpler, and it is certainly more spontaneously probable—vastly more probable—and unless you believe in rain gods, is not likely to be perceived as purposeful.

Three things set screwdrivers, viruses, membrane proteins, and action potentials apart from water waves, solar systems, and thunderstorms: prespecification, purpose, and the appearance of extremely low probability. Instruction use accounts for all three. Instructions, by their nature, have the purpose of prespecifying something, and the amount of specification determines how improbable something would be if instructions were not used. The laws of nature are also specifications of a sort, but they are simple, universal, and never focused on particular outcomes. In contrast, instructions are complicated, not universal, and lead to outcomes that are *for something specific*. Instructions are always motivated; they represent investment in an outcome. Things resulting simply from the laws of chemistry and physics acting on reasonably probable initial conditions may be surprising, intricate, or grand, but if we exclude religious interpretations, can never be said to be designed for a purpose or predetermined to fit something else.

Screwdrivers and viruses cannot be explained solely in terms of simple rules acting on unexceptional initial conditions. Focused special rules (instructions) must be employed to establish conditions necessary for a creation that otherwise would be too improbable to come about. When structure occurs as the result of instruction use, its apparent improbability flows directly from the improbability of the instructions required to make it. Since all but the shortest instructions are wildly improbable if generated randomly, we know with certainty that the information characterizing them did not arise by some random coming together of bits. Notions of self-organization do not free us from this conclusion. Far from equilibrium phenomenology

opens up whole realms of dynamic possibility, but does not solve the improbability problem and does not explain purposefulness. Only instructions do that.

If we set aside human creativity for the moment, the only known strategy for creating purposeful bodies of information, which instructions certainly are, is a probabilistic computation that is naturally carried out by populations of replicating entities subject to repeated cumulative selection. I call this computational strategy the engine of complexity. It is this same computational strategy that underlies biological evolution and several other phenomena we will discuss in later chapters. In the absence of cumulative selection, there is simply not enough time and space in the universe to find a particular combination of symbols that encodes purposeful instructions needed for a particular outcome. Selection is nonrandom, and selection rules provide a compass that allows the identification of useful combinations of information from infinitudes of non-useful alternatives. This is true even if the bits of information being selected are produced randomly. Structures that exist because they survived selection always have a kind of purpose; they fit the criteria that determined their continued existence better than those that were not selected. Instruction-based objects and behaviors are purposeful because their instructions were shaped by the rules active during repeated selection.

This is not to say there is no reason that water waves and thunderstorms exist; it is just that the reasons are direct and immediate. They are spontaneous responses to ordinary conditions. Water waves, thunderstorms, and sandpile avalanches are all mechanisms for transporting energy. To understand them, we need to know about physical forces and impediments to those forces operating in well-defined systems. Impeding energy flows leads to potential energy buildups that sometimes manifest as structure. The rest is just details. Structure for free in our universe is very important, but alone does not explain our existence. For this we also need the concept that information, appropriately organized, can direct the formation of highly improbable purposeful objects. Without such extra information, nature is limited to exploring the most probable of an infinitude of potential structures. However, *by using appropriate instructions any logically permissible structure is attainable*—no matter how improbable it

might seem (assuming the needed resources are available). This is a pretty remarkable statement.

Purpose implies need. Something is created so that certain criteria will later be satisfied. When applied to Bénard cells this concept is empty. Bénard cells form because coordinated movements of molecules happen to most efficiently transport energy from hot to cold; there is no advance plan. When a pan of water is placed on a burner, a temperature difference is established and the mechanism that best transports heat from bottom to top of the water column is the motion we call convection. When the geometry is just right, convection occurs in a regular pattern. The laws governing heat transfer were not created so that Bénard cells would form.

Viruses transform mechanical and chemical energy into heat and thereby participate in the flow of energy through the biosphere, ultimately contributing to the flow of energy from the sun to outer space; but viruses are hardly the most efficient mechanisms for doing this. The point of their existence is not energy transfer, but perpetuation of viruses—something quite different. Viral structures are not motivated by the relief of an energy potential, but by a drive for continuation. This drive utilizes energy potentials, but is not motivated by them. Proteins can be understood in the same way. They are tiny aspects of vastly complex structures we call cells that are also motivated by a drive to continue. Proteins can be seen to have the purpose of enabling a cell or virus to exist, or a body to live; and all living things have the purpose of continuing the genetic information that makes them what they are. There are uncountable numbers of possible proteins; those that actually exist are special because they contribute to the continuation of living organisms. To learn more about the drive of genes to continue on, one should read *The Selfish Gene* by Richard Dawkins.<sup>6</sup> A drive for continuation is a central feature and a natural outcome of the engine of complexity in operation.

Other things being equal, we expect the structures we encounter to be reasonably probable given local conditions and history. This is why someone from another planet would see screwdrivers as indicative of intelligent life. Screwdrivers are not probable except in a world where people use screws to build things. They exist because of a human need (goal) for turning screws. Screws, in turn, exist for the purpose

of fastening things together. The chain of motivation may be long, but it cannot be ignored if we are to understand purposefully complex objects. Solar systems and mountains also result from long chains of events, but they are chains motivated by physical forces and expediency, not by need. The challenge is to understand how needs get translated into goals and goals into structures satisfying those needs. This is the subject of the next chapter.

---

## Improbability and the Engine of Complexity

How do order and disorder relate to each other?

Remember predigital television sets? I still use one. When I turn to a channel having no nearby transmitter, the screen exhibits “snow,” a random flickering of pixels. Dialing to another channel, I see actors performing or scenes of faraway places. The difference between disorder and order on the screen is clear. The snowy screen has the same number of pixels as the screen showing a talking head, but they are arranged differently. Salt crystals, viruses, and action potentials provide examples of order found in nature; gases and liquids lack structure. When the parts (pixels or molecules) have relationships to one another, we recognize structure. Structure is a manifestation of order, and although order and structure are commonplace in our experience, they are trickier to define than you might guess.

Coin tossing provides a good illustration. The expectation is that on repeated flips, a head will come up 50 percent of the time, and we say the odds are 50/50 that heads will result from a particular flip. The expected outcome can be skewed by various tricks, but if you don’t use



a biased coin, the result of 1,000 flips will be close to, though rarely exactly, 500 heads and 500 tails. It seems reasonable that if you flipped a coin 1,000 times and wrote down H for head and T for tail, the resulting string of 1,000 Hs and Ts would constitute a “random,” structureless, one-dimensional arrangement of the letters. But is this always true? Does random generation guarantee random order?

The answer is no. When the coin-flipping experiment is repeated many times, occasional outcomes exhibit an obvious pattern. If you flip one head, the odds the next flip will be a tail is 50/50 (probability  $1/2$ ). If you have already flipped a head and tail in that order, the odds of the next flip showing a head is also 50/50. So, the probability of flipping a head, a tail, and a head (H·T·H) in that order is  $1/2 \times 1/2 \times 1/2$ , or  $1/8$ . The probability of flipping H·T·H·T·H·T·H·T·H·T is  $1/1024$ . In principle, possible outcomes of 1,000 flips include: 1,000 alternating heads and tails, two heads followed by three tails (H·H·T·T·T) repeated 200 times, 500 heads followed by 500 tails, and even 1,000 tails with no heads ever being tossed. Though these orderly outcomes are formally possible, we should never expect to see them happen. This is because the probability of any particular outcome of 1,000 coin flips is  $1/2^{1000}$ , an unimaginably small number. We could flip coins for the life of the universe and never see a particular preselected outcome.

Coin tossing illustrates a principle that applies to all objects (or systems of objects) made of parts: random placement of those parts usually but not always results in disorder. This principle applies to linear sequences of symbols, to the pixels making up a TV screen image, and to three-dimensional arrangements of molecules. Analysis reveals that the larger the number of parts, the smaller the fraction of possible configurations of those parts that will be nonrandom. At one extreme, all arrangements of Hs and Ts that are two-symbols long seem orderly, consisting either of the same symbol repeated twice or of alternating symbols. At the other extreme, the number of infinitely long sequences that are orderly is mathematically zero.

A challenge for this sort of discussion is that perceived order is subjective. Fortunately, there is a rigorous way to define randomness and nonrandomness, and it comes from computer theory. According to this way of thinking, a random sequence is one for which there are no internal relationships, that is, no correlations, between elements of the sequence. We considered this idea in chapter 2. If the

elements are, for example, letters in the English language, then a random sequence of letters would be one for which no procedure other than copying exists that would allow prediction of a particular letter in the sequence even if one is given all the other letters. A nonrandom sequence, one exhibiting internal relationships, is characterized by rules relating different parts of the sequence. An example is provided by 1,000 As followed by 1,000 Bs. A simple rule exists that relates the As to the Bs. The rule “write 1,000 As followed by 1,000 Bs” will produce the desired sequence.

In computer science, the input to a computation is considered to be a description of the output, and a defining difference between ordered and disordered objects is that ordered objects admit short descriptions. In the case of very short sequences, such as H·T, a simple rule exists relating the parts, but the rule is longer and more complex than the sequence itself; hence we can’t tell whether or not such a sequence is orderly. For H·T·H·T·H·T·H·T·H·T the rule “repeat H·T five times” is slightly shorter than the actual sequence if we write it in a compact form such as “5 × HT.” For 1,000 alternating Hs and Ts the rule “1,000 × HT” is *much* shorter; so we clearly recognize this sequence to be ordered.

A not particularly intuitive aspect of computing is that most possible outputs cannot be produced from input shorter than the output. This means that the overwhelming majority of long sequences are random. If internal relationships are present in an output, then inputs shorter than the output are possible. “Interesting” computational objects are almost always found among the small fraction of configurations that have extensive internal relationships—that is, those that have structure. Sometimes the order present is too subtle for the human eye to recognize, but if an object (an output) can be created by deterministic computation from short input, we know some form of order is present in it. This provides a quantitative definition of order: *an object possesses order if a description of it is possible that is shorter (smaller) than the object itself.*

Another important point about randomness in objects is that order and randomness are not all-or-none properties. An object can be more or less orderly. In fact, most objects possess both order and disorder. A long sequence of Hs and Ts produced by tossing a biased coin might be 55 percent Ts and 45 percent Hs. Such a sequence would not be

highly ordered, but still it would exhibit some regularity. Our bodies have a lot of structure, but also a lot of randomness at the cellular and subcellular levels.

To summarize where we are in this long argument, orderly arrangements are characterized by internal relationships, and an important property shared by all physical systems is that the overwhelming majority of imaginable configurations exhibit no internal relationships. For large systems order is unexpected and calls out for explanation.

## How do we reconcile great improbability with existence?

A quote from Douglas Adams' book *The Hitchhiker's Guide to the Galaxy* gives us a fun starting place:

Guide: The Infinite Improbability Drive is a wonderful new method of crossing interstellar distances in a few seconds; without passing through all the tedious mucking about in hyperspace. As the Improbability Drive reaches infinite improbability, it passes through every conceivable point in every conceivable universe almost simultaneously. In other words, you're never sure where you will end up or even what species you'll be when you get there. It is important to dress accordingly.<sup>1</sup>

In his fanciful story, Adams' character Arthur Dent was ejected into deep space without a space suit and rescued from certain death by the *Heart of Gold*, a remarkable craft that moves from place to place by means of an "infinite improbability drive." The improbability of Arthur Dent's rescue was given as "two to the power of two hundred and seventy-six thousand to one, against" ( $1/2^{276,000}$ ). Every reader recognizes the ridiculousness of this, but why? An answer requires that we explore what we mean when we say something has a probability of existence.

We can start with the imaginary space of all possibilities. Existing things occupy a tiny fraction of it. Changes of any kind constitute movements to new locations in the space. In the absence of rules, a change would constitute a jump to a new randomly chosen location, very reminiscent of the *Heart of Gold*. The problem with this concept is that the overwhelming majority of all unrealized objects in this

imaginary space are random. Because it is so big, the fraction of the space characterized by orderly things like you and me, solar systems, and skyscrapers is vanishingly small. A *Heart of Gold* jump to a randomly chosen place in the space of all possibilities would not be to another orderly state but to a state without any order at all.

Structure only exists because our universe operates according to nonrandom laws and has a specific history. Without rules, orderliness is far too improbable. In the physical universe, structure comes about because laws of chemistry and physics restrict possible outcomes and because outcomes are built on previous outcomes already exhibiting structure. At one level, physical laws “explain” why our universe exhibits structure, and at another, the origin of these rules is the deepest mystery of our existence. It is important to remember that while the laws make order possible, they do not ban randomness; after all, the outcome of a coin flip is randomly heads or tails.

For 400 years scientists have observed the universe, deduced the rules in play, and performed experiments to test their deductions. The rules determine what paths are permitted through possibility space because they disallow many events, such as turning a black hole into a silver dollar or a silver dollar into Arthur Dent. Permissible events create conditions for other permissible events, establishing allowed paths. Within permissible paths, many situations allow random actions to occur, and sometimes they are the norm. To give an example, physicists know that every atom of uranium 235 will decay into thorium 231 plus an alpha particle, but they don’t know when the decay will occur. The probability for a particular atom is 50/50 for the conversion to occur sometime in the next 700 million years. The likelihood of a particular uranium 235 atom decaying in our lifetime is very small, but not zero. A small block of uranium metal is radioactive because it contains so many atoms that a few decay every second. The timing of that existence-changing event (for the atom) is quite random. The phenomenon can be described by an equation, but only as a statistical property of many atoms or as a probability of decay.

When rules are followed, a property of paths over time through the space of all possibilities is that future states are contingent on past events. Structure builds on past structure. In another example involving radioactive decay, phosphorus 32 is an unstable isotope of the element phosphorous. Phosphorous is an integral part of the structure of

DNA, and the common isotope phosphorus 31 is completely stable. When an atom of phosphorus 32 happens to be incorporated into DNA, all is fine unless it decays. If that happens, the DNA strand is broken. If a phosphorus 32 atom is incorporated into a gene that codes for an enzyme needed for the synthesis of melanin (the pigment that bestows dark color on mammalian skin and hair) there will be no effect unless it decays and damages the DNA. If the decay should occur in a sperm or egg cell nucleus, a resultant embryo would likely experience a defect in melanin synthesis. Usually such mutations are recessive and they don't show immediately, but if in a future descendent two mutant genes are combined, then the individual will be albino. If the organism is a wild rabbit, it will likely be eaten at an early age because unless the ground is snow covered it cannot hide easily from predators. If the organism is a human being living in a primitive society, he or she may be greatly honored and attributed with magical powers. On the other hand, if the decay occurred in a sperm cell that did not participate in fertilization, there would be no biological effect at all. The difference between no effect, being eaten, or having exalted status is determined by specific events in an historical path. When sequential processes admit random events, the possible future becomes different every time such an event occurs. Looking backward in time, the current state is seen to be contingent on the particular sequence of random and non-random events that occurred in the actual history. If the system could be rerun, the outcome would be different because the chance events would be different.

When chance events are combined with selection, the resulting outcomes are nonrandom. This may seem odd because random changes are usually seen as destructive to existing nonrandom structure. But when many random changes are considered, a few actually contribute to organized structure. When random changes are small and occur in a system that offers parallel opportunities for acceptance or rejection, and when the choices made are based on consistent criteria, then the surviving structures inevitably come to reflect the criteria that decide which are saved and which are not. The surviving structures can become more and more organized and better at whatever they do. Because of this, evolution should be seen as a nonrandom method for finding new paths through possibility space even when the initial changes to be selected are randomly generated.

The full power of repetitive selection is only realized when it is applied to instructions. Recall that appropriate instructions make achievable *any structure* allowed by chemistry and physics (if you have the necessary resources). Because repeated selection is able to accumulate useful information (from the perspective of whatever forces are responsible for selection), and because over time selection continues to operate on the outcomes of ever-better instructions, those outcomes may eventually seem to be as apparently improbable as anything achieved by the infinite improbability drive.

### Where do instructions come from?

The key to understanding the existence of things that seem hopelessly improbable when viewed strictly in the context of chemistry and physics is recognizing that information can be organized and used to specify otherwise hopelessly improbable structures. Information when utilized in this manner changes outcomes from impossible to easy. This observation requires that we focus our attention on the origin of organized information possessing such wonderful qualities. The instruction “write one thousand Hs in a row” enables creation of an object that is effectively impossible to achieve by recording random coin flips. Chapter 4 explored several structures that are only possible because instructions are used in their formation. Many, but not all, instructions are products of human ingenuity, but human brains, special as they are, cannot create purposeful information out of nothing.

An opportunity made possible by the nature of computation is having the output of one computation serve as the input to another. This establishes the possibility of long chains of computations with the output of each serving as input to the next. A variation on this theme is for the output of a computation to serve as input to the same computation (same rule set). Such a process is diagrammed as a cycle in figure 5.1. We say it is iterated (repeated over and over).

Most computations cannot be iterated because their outputs are not structured to serve as inputs; but it is not difficult for a programmer to design an iterated computation. If iterated computations are deterministic, they have three possible fates: they converge over time to some final unchanging output/input, they enter a cycle where the same

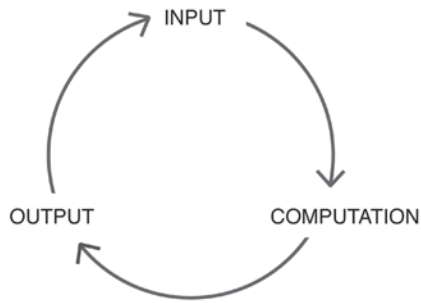


Figure 5.1. Diagram showing the concept of iterated computation.

output/input occurs over and over at regular intervals, or they eventually stop. If the computation is probabilistic, a fourth possibility arises: the outputs may continue changing forever.

A type of iterated computation that is easy to implement is iterated copying. Clearly, if some structure can be copied, then a copy of that structure also can be copied. Iterative computation of the form *input* → *copy* → *input* → *copy* . . . is not very interesting, but it works, a fact easily demonstrated by repeatedly placing the output of a copy machine in the “in” tray. When the copying process is less than perfect, successive outputs will slowly degrade over time, becoming more and more disorderly. Disorder due to the introduction of errors represents the loss of meaningful information. Information loss is typical of iterated imperfect copying without selection. But, couple iterated copying with selection, and everything changes.

Programmers who create iterative computer programs usually divide the input into an unchanging part and a changeable part. Things are easier to manage when the rule part or algorithm is not allowed to change and only the data evolves. Invoking the example of imperfect copying by copy machine, only the copies change over time, not the machine itself. If the rule or algorithm part of a program is also subject to unplanned change, then there is a high risk of creating invalid programs resulting in a stopped cycle (the altered machine no longer works). Systems with this property are difficult but not impossible to implement in electronic computers. In living systems, the program, machine, and data are intertwined, and all change over time. Only the logic of the evolutionary process remains the same even as the means of carrying it out evolves.

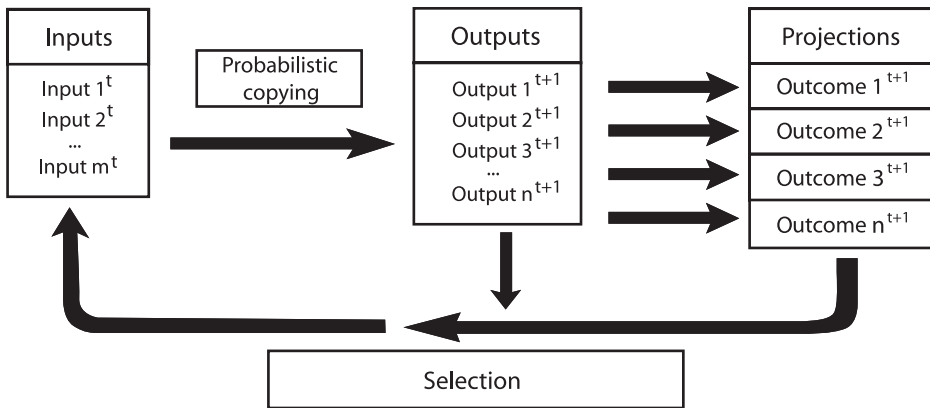
There are general strategies that allow both algorithm and data to be subjected to random change while assuring the cycle continues. Three features are generally present if the possibility of program change is to be reliably incorporated into iterative computation. These are:

- the program must specify some sort of mechanism for copying, and this mechanism must allow for errors or other changes to occur
- errors (changes) made in any one cycle must be small
- there must be a sufficient number of outputs in each cycle so that at least one of the outputs is not significantly changed from the current input to that cycle.

If these three features are present, a long-lasting iterative computation can occur in which the program (rules and input data) changes over time, and the computation does not stop. The provision for multiple outputs in each cycle demands that some sort of selection be in operation because no system can maintain a population of inputs/outputs that grows forever larger without limit. Let us call the general computational strategy that encapsulates these features “iterated probabilistic computation with selection” (IPCS). Such systems tend to exhibit progressive change. The essential features are diagramed in figure 5.2. The inner cycle is the defining feature of all evolutionary processes and is what I call the *engine of complexity*. The engine is an IPCS. Because of the multiple inputs and outputs, it is a parallel computation.

Any system that incorporates the engine of complexity cycle will normally accumulate information in the form of input/output structure that improves entity chances of being copied again. The cycle extracts information from whatever random changes are made by retaining any change that better satisfies the selection criteria and rejecting any change that does not. I call such accumulated information “purposeful” because it exists to satisfy the selection criteria. Every time a probabilistic change is made, the resulting output is evaluated according to the selection rules. If a change is favored, it becomes input to the next cycle, if not, it is discarded. In biology the selection process is messy and probabilistic. Some positive changes that occur are never passed on to the next generation and some deleterious ones are, but on average the population of entities comes to incorporate many of the better changes, and most of the deleterious ones are eliminated. An IPCS





*Figure 5.2.* Diagram of iterated probabilistic computation with selection (IPCS). This computational strategy is the engine that underlies all evolutionary processes. The superscripts  $t$  and  $t+1$  indicate the cycle number and  $m$  must on average be less than  $n$  (selection must occur). Both inputs and outputs encode information.

computation is the informational equivalent of a noisy mechanical ratchet, inching forward most of the time. The requirement that at least one of the outputs is not significantly changed (bullet 3, listed earlier) assures that the system as a whole does not slip back. Living systems generally satisfy this requirement by having a sufficiently low mutation rate so that in each generation most individuals are not significantly less fit than their parents. Unusual environments with high radiation levels or that provide high exposures to chemical mutagens can create situations where the mutation rate is too high and the requirement is not met. Such populations enter a spiral of decline where each generation is less fit than the previous one.

The boxes labeled “projections” in figure 5.2 are not an essential part of the engine, but are shown for clarity because most evolutionary systems have this feature. In such systems selection acts on outcomes and only indirectly on outputs. In biology outputs are DNA sequences, and outcomes are organisms. In the sphere of human activity, outputs are instructions that people follow, and outcomes are artifacts or social structures specified by those instructions.

It is important to reemphasize that changes introduced in each cycle must be small. The larger a change made to an input, the more different

the resulting output and outcome will be. But, this also means the less likely it is to well satisfy the selection criteria. Large changes produce large jumps in possibility space and, accordingly, encounter the probability problem. In the natural world, structures in “nearby” possibility space are generally more similar to one another than to structures farther away; because of this, there is a reasonable probability that some structure nearby to the current structure might be an improvement. In contrast, structures “far away” in possibility space are always very different. And different is usually bad because far away structures are almost guaranteed to be more disorderly and measure up poorly against whatever quality is being assessed. To illustrate this, imagine an airplane. A small change might be adding one foot to the length of the wing. A larger change would be changing the number of wings. Still larger changes would be changing the overall shape to that of a boat or a large metallic boulder. Only small changes are likely to be compatible with flying, and only some of them might constitute an adaptive improvement on the original. Given this, the only practical way to find good solutions is to inch toward them, small step by small step. Imagine traversing a zigzag catwalk in the dark with no hand rails. Only by taking very small steps can a walker avoid falling off. Large steps are counterproductive.

Probabilistic computations by their very nature introduce random changes. Every time a probabilistic computation is performed, at some probability changes are made. In many systems, the mechanism is simple error. Biology and some computer programs utilize more complicated change mechanisms that include rearrangements and the mixing and matching of previously successful sequences. Regardless of the details, for an engine of complexity computation to operate, at least one of the  $n$  outputs in each round (figure 5.2) must be suitable to serve as input to the next cycle. If none do this, the computation stops. In biology no individual would reproduce and the population would go extinct.

Selection can be based on any rule or combination of rules that non-randomly picks which outputs will serve as inputs to the next cycle. In nature selection involves interactions between outcomes (organisms) and the environment. Sometimes the selection rules are explicit, such as “dead animals do not have offspring.” Other times they can be indirect and probabilistic, such as “maybe a predator is hiding behind

that rock.” Regardless, a system featuring the engine of complexity will adapt over time responding to whatever nonrandom selection rules are in play. As it adapts, the system naturally accumulates information pertinent to satisfying the selection rules. If new selection rules are imposed, the system will immediately begin accumulating information pertinent to the new rules. The source of the new information is random change selected. The simplest possible choices made and saved constitute one bit of information. By making repeated simple choices and recording them, a system can build up pertinent information bit by bit. When selection is nonrandom, the resulting accumulated information is also nonrandom.

For completeness I need to make clear that the engine of complexity does not require random change. The basic cycle shown in figure 5.2 works with any source of change. Because the spectrum of possible outcomes is restricted whenever the range of possible changes is restricted, the creativity of the process is reduced by predetermining the changes made. Random change offers maximum opportunity, but when a desired outcome is known, dictating specific changes can dramatically speed up the process. The beauty and power of the IPCS strategy is that it exhibits maximum creativity in the absence of preplanning.

## How do instructions change our view of object probability?

Once you accept the central role of instructions in the formation of complex things, it is natural to view the probability of objects in terms of the probability, or should we say improbability, of the instructions required to make them. Complicated instructions allow complicated physical objects. The specialness of objects that are made according to instructions flows directly from the specialness of their instructions. To appreciate how truly improbable long instructions are, we need to visit the realms of large and small numbers.

Human use of numbers originated with the invention of counting. Some primitive tribes never invented counting beyond two.<sup>2</sup> From this we may infer that the ability to count is not built into our genes. Individuals in such cultures cannot consistently exchange a bowl of rice for a dozen oranges. They can approximate twelve oranges, but they cannot get it right all the time without laboriously comparing twelve oranges

with twelve more oranges. Counting makes commerce practical, and a merchant who knows how to count can easily cheat a merchant who does not. Not long after the invention of counting, someone realized that she could also count fractions of objects. Thus, if she cut an apple into four parts, each quarter apple also could be counted.

A simple way to count beyond the number of our fingers and toes is to associate a name with each number. The ancient Egyptians had a unique name for each number between one and one thousand. Such a system works if you are dealing with moderately small numbers of objects and do not care to manipulate numbers, but numbers are simpler to keep track of if some sort of repeating system is used. For example, in the English language we use the composite phrase one hundred and twenty-three instead of having a simple word for this number. But composite word use also gets cumbersome, and replacing words with symbols saves a lot of space. So, instead of one hundred sixty-two thousand, six hundred and seventy-four, we can more conveniently write 162,674. More important, symbol-based counting facilitates number manipulation. The decimal system is particularly convenient for arithmetic. If you believe arithmetic is equally easy in all number systems, try dividing Roman numeral LIV by XVI.<sup>3</sup> Ease of arithmetic explains why the decimal system is so widely used in the world today.

Another useful feature of the decimal system is that it can represent numbers that are too large or too small to ever be encountered in our everyday world. Thus, as I write this sentence, the number of human beings on the planet Earth is estimated to be seven billion, seventy-two million, five hundred and seven thousand, three hundred and two, or 7,072,507,302 (and increasing at the rate of three per second). The accumulated deficit of the U.S. government on October 13, 2012, at 10:54 A.M. Central Daylight Time was \$16,168,864,374,347. With modest training, anyone can read and write such numbers even though what they represent is beyond our actual experience. To see this, consider that if you meet twenty new people a day for seventy-five years, you would meet about half a million (500,000) people in your lifetime. If you set about counting dollar bills at the rate of one per second, you could count \$36,000 in ten hours (assuming that you did not lose count). In a year the count would be up to \$13,140,000 and in a lifetime, approximately 1 billion dollars. To count 16 trillion dollar bills would take 16,000 lifetimes. To meet everyone in the world at the rate

of twenty per day would take more than 13,000 lifetimes—assuming that no one new was born and no one died during the count. These numbers are tiny compared to some other numbers we might encounter. Table 5.1 provides a range of examples.

The standard decimal system also has its limitations. Very big numbers (and also very small numbers) are time consuming to write and hard to read. It is preferable to use a shorthand known as scientific notation. In scientific notation, 1,000,000 becomes  $1 \times 10^6$  (or just  $10^6$ ), 3,400,000 becomes  $3.4 \times 10^6$ , and 0.0002 becomes  $2 \times 10^{-4}$ . The superscript 6 indicates six zeros (or digits) to the left of the decimal point and to the right of the first digit; and the superscript  $-4$  indicates four digits, the last of which is nonzero, to the right of the decimal point. Using this notation it is easy to write a number like  $6 \times 10^{5000}$ , which to communicate would otherwise require a six followed by two pages of zeros. Table 5.1 uses scientific notation.

The first and last entries in the table come from the physical theory of gases. The second number in the table is motivated by the observation that this book has roughly 500,000 characters and that about forty different characters are used (twenty-six letters, a space, and various symbols such as /;–"). There are  $40^{500,000}$  ways to arrange 500,000 symbols taken from an alphabet of forty symbol types, and  $40^{500,000}$  is the same as  $1 \times 10^{800,000}$ . This is the number of different books of exactly 500,000 characters that can be written using any particular alphabet of forty symbols. Of course, almost all such books would be sheer gibberish; just a few would have meaning to someone who understands the language being used. This also shows how special an intelligible book is.<sup>4</sup>

An advantage but also a drawback of scientific notation, sometimes overlooked, is that it becomes too easy to write really big and really small numbers. In science numbers sufficiently large or small can be wildly misleading. In quantum mechanics one cannot specify both the position and momentum of a particle with limitless precision. If you try to define a volume of space that is infinitesimally small, the uncertainty about what is going on there becomes enormously large. Because of this, at very small dimensions, all existing physical theory breaks down, and hence scientific understanding of what happens at these scales also breaks down. The linear dimension at which quantum mechanics fails is around  $1 \times 10^{-35}$  meters (Planck length). Because

Table 5.1

Some big and small numbers.

Number of distinct molecular states of one liter of gas	$1 \times 10^{50,000,000,000,000,000,000,000}$
Number of possible books with exactly 500,000 characters	$1 \times 10^{800,000}$
Atoms in the universe	$1 \times 10^{80}$
Stars in the universe	$1 \times 10^{22}$
Age of the universe in seconds	$4 \times 10^{16}$
U.S. national debt in dollars	$1.6 \times 10^{13}$
Age of the universe in years	$13.7 \times 10^9$
Number of human beings living on Earth	$7.1 \times 10^9$
Number of people who live in Ames, Iowa	$5.2 \times 10^4$
Probability of rolling two 1s with two dice	$2.7 \times 10^{-2}$
Probability of dealing a royal flush from a 52-card deck	$1.54 \times 10^{-6}$
Probability of being killed by a meteorite	$1 \times 10^{-10}$
Probability of any particular sequence of 50 coin flips	$1.1 \times 10^{-15}$
Planck length, the smallest length that makes sense in physics, in meters	$1 \times 10^{-35}$
Probability of one particular sequence of 1,000 coin flips	$1 \times 10^{-300}$
Probability of your DNA sequence occurring by chance	$1 \times 10^{-2,000,000,000}$
Probability of a particular molecular state in one liter of gas	$1 \times 10^{-50,000,000,000,000,000,000,000}$

the fastest that information can be transmitted is at the speed of light ( $3 \times 10^8$  meters per second), it takes light at least  $1 \times 10^{-44}$  seconds to traverse  $1 \times 10^{-35}$  meters. Hence, to speak of times less than  $1 \times 10^{-44}$  seconds is meaningless in a scientific context. A school child can easily write a time of  $1 \times 10^{-1000}$  seconds, but to what purpose?

Other considerations apply to large distances and long times. The latest astronomical evidence combined with theoretical models provides strong evidence that the age of the known universe is about 13.7 billion years. The defining event, the big bang, marks the beginning of

time, so past times greater than  $4 \times 10^{17}$  seconds (13.7 billion years) are undefined. What is the meaning of time before time? Likewise, because light travels with finite speed, astronomers can never see objects that are so faraway that light from them would need to have traveled more than 13.7 billion years to reach us. One can speak of greater distances, but one must be very careful when referring to something that cannot ever be detected.

Returning now to probabilities, the odds of a fair coin toss resulting in “heads” is  $1/2$ . If we roll an honest six-sided die, it will come up “two” one-sixth of the time. Most card games are based on probabilities, and using a well-shuffled fifty-two-card deck the odds of dealing out an ace, king, queen, jack, and ten of the same suit in a five-card hand is  $1/649,740$  (probability of  $1.539 \times 10^{-6}$  in scientific notation). Unless you have played a lot of cards, you have probably never seen a royal flush honestly dealt. Your odds of dying as a result of an airplane hitting you while you are on the ground are four in a million (probability of  $4 \times 10^{-6}$ ) and of a (small) meteorite striking and killing you, roughly one in 10 billion (probability of  $1 \times 10^{-10}$ ). As you might expect, with more than 7 billion ( $7 \times 10^9$ ) people, hundreds have been hit and killed by airplanes, but there is only one credible report of a person being killed by a meteorite. Even though there is a quantifiable danger, few people worry when they go to bed at night about being hit by an airplane or a meteorite. The odds of these things happening to any particular person are too low to fret about—even though they could happen.<sup>5</sup>

Though very small, these probabilities are comprehensible. As we consider sequential random events, however, when the number of events becomes large the probabilities of particular sequences happening become incomprehensibly small. And when probabilities are sufficiently small, it is actually misleading to treat them as somehow different from zero. ***A sufficiently small probability means that the event in question has never happened and will never happen through a random process,*** even though its calculated probability is ever-so-slightly greater than zero.

Sequences of letters provide easy-to-understand illustrations of such incredibly small probabilities. Consider again consecutive coin flips. The outcome of each flip is either a head or a tail (the probability of either event being 0.5). Sitting at my desk I performed 50 such flips

and the outcome was: H·H·T·T·H·T·H·H·T·T·T·H·T·H·T·H·T·H·  
H·T·H·H·T·H·T·T·H·T·H·H·T·T·H·H·H·H·T·H·H·T·T·T·H·T·  
H·T·H·H·H·T.

There are  $1.1 \times 10^{15}$  ( $2^{50}$ ) possible sequences fifty letters long that consist of only H and T. So, the probability of this particular one is roughly  $1 \times 10^{-15}$ . If everyone on earth flipped a coin every second and the results were recorded, the above sequence would reoccur somewhere in about a day. If I had recorded 100 flips instead of fifty, it would take all of humanity doing nothing but flipping coins about  $5 \times 10^{12}$  years (400 times the current age of the universe) to reproduce that particular sequence! A sequence generated by 1,000 flips has a probability of  $1 \times 10^{-300}$  and would take  $5 \times 10^{282}$  years of flipping to reoccur. Such a probability is indistinguishable from zero. If all the computers on earth were doing nothing but simulating coin flips for a million billion zillion times the lifetime of the universe, they would have no chance at all of generating that one particular sequence. To put this another way, if you were to write down a nontrivial sequence of 1,000 Hs and Ts, ***you can be absolutely certain that that particular sequence will never again occur by a method based on chance.*** This observation demonstrates an obvious but nevertheless important relationship between probability and existence. If a hypothetical object's probability is sufficiently low, it *will never* come into physical existence by chance alone. Remarkably, whenever a low-probability object does somehow exist, we can make copies of it without reencountering the probability problem at all. Clearly, probabilities calculated on the assumption of random assembly can be very misleading.

Modern human society routinely creates and uses symbol sequences very much longer than 1,000. Most instructions for making useful things are comprised of long symbol sequences; books and compact discs store them. All these sequences are very special. The probability argument tells us we can be absolutely certain that no meaningful long instruction ever arose through the random arrangement of symbols. So, how do long instructions come about? Saying that a person created them does not solve the problem, since the mechanism a person uses to do so is unknown.

We have already seen a potential solution. The engine of complexity computation is quite capable of creating large purposeful orderings of symbols (instructions are an example of such purposeful orderings)



when the desired ordering is not known in advance. In fact, *the engine of complexity is the only known mechanism capable of doing this efficiently*. The implications of this for understanding how humans think things up and technologies get created cannot be ignored if we really want to understand how complicated things come about.

The next chapter explores in detail how one particular computer implementation of the engine of complexity solves a simple “one-max” problem. The task is to create a string of all 1s without programming the answer into the input. You can do this easily because you “recognize” the difference between a 0 and a 1. To get a computer to do it with telling it how to “recognize” the difference is a little trickier. One way is to program the computer to randomly generate strings of 0s and 1s until a string appears whose digits add up to the length of the string. The problem with this method is that to generate a string of one thousand 1s this way requires the creation of approximately  $10^{300}$  different strings. This cannot be done in much less than  $10^{100}$  years in a universe of only  $10^{80}$  atoms (limiting the number of computers to work on the problem). Remarkably, by invoking the engine of complexity the problem is solved in less than one minute on my desktop computer—and the efficiency gain provided by the algorithm gets increasingly better the longer the string! In our everyday experience, instructions much longer than 1,000 bits are common. It is beyond inconceivable that they were generated by random means, and the only known scientific alternatives are: (1) the universe is predetermined to produce all the instructions ever used in the past, present, and future; or (2) they are generated by means of an efficient computational strategy. The only one we know of capable of assembling large purposeful bodies of information when the desired informational structure is not known in advance is what I call the engine of complexity. Saying that someone “thought up” a long set of instructions just obscures the issue. “Thinking up” is not a mechanism for it tells us nothing about what computationally goes on in the brain.

In this section we have looked closely at the random creation of things to drive home the point that *for complicated things, random creation is not an option*. This may seem obvious, yet it is a very important principle, in part because it can be turned around. If it can be established that random assembly of an object is impossibly improbable, then we know it was made nonrandomly. This was Paley’s argument,

but when he made it he didn't know about the engine of complexity; God was his only alternative.

Philosophically, what does nonrandom assembly mean? From one point of view, creating something nonrandomly means that *prior knowledge* plays a role in its creation. When we write a letter or an essay, we are able to construct specific sequences of letters with specific meanings because we already know something about what we are writing. If we knew nothing, we couldn't write a letter. A chimpanzee can be taught to use a keyboard, but the resultant strings of letters from its typing will never have meaning to a human being. By the same token, scientists will never create new life forms in a test tube by synthesizing new DNA sequences and expressing them in cells until they know in advance which sequences to synthesize. Random sequences won't do.

From a different point of view, nonrandom creation means that *constraints* are placed on possible outcomes. Constraints characterize the physical world. Consider a picket fence. The upright boards of a well-built fence are all vertical, evenly spaced, and the tops are all the same height above the ground. Why are so many fences like this? After all, a large number of fence designs are possible using the same pickets, and most possibilities are very disorderly. The reason for the observed order is builder-imposed constraint. Each picket is nailed in place only when its position meets certain criteria. The laws of chemistry and physics constitute constraints just as much as fence builders' rules, except that the laws of chemistry and physics apply to everything. Selection is also a constraint, and repeated selection builds up nonrandomness. Instructions are a special kind of constraint, and the longer (and more complex) the instructions, the more complicated can be their outcomes. Thus, the engine of complexity by assembling instructions provides a way for complex things to exist, and it does this by assembling the necessary information in a methodical, nonrandom way. Without the engine much of what we see and use would be hopelessly improbable. With it, wonderfully surprising things become commonplace.

### Is life the outcome of a computation?

The evolution of life is the best-studied and most famous example of the engine of complexity in action. When Darwin published his

famous monograph *On the Origin of Species by Means of Natural Selection* in 1859,<sup>6</sup> he knew nothing of the modern concepts of information, computation, proteins, or DNA. He didn't even know about genes. He based his theory on careful observation and logical thinking. A keen observer of nature, he was greatly impressed by the variation observed in natural populations and also by the fecundity of nature. In every natural population, many more offspring are produced than survive to reproduce. How, he asked, could evolution *not occur* if there were even a slight bias favoring the characteristics of some offspring over others?

A testament to his powers of observation and sound logic is that all the subsequent discoveries in genetics, biochemistry, molecular biology, cell biology, and computational science have strengthened rather than weakened the case for the theory of evolution. This is not to say that there are no detractors. A few scientists and many more nonscientists argue that the theory of evolution fails because it leaves some important issues poorly explained. Foremost among these is the ultimate origin of life but also the origin of major innovations such as flight, vision, and the vast complexities of cellular biochemistry. All scientists who have studied the issue accept that the origin of life is poorly understood. Darwin himself made clear that his theory did not explain the origin of life, and additions to the theory over the past 150 years have not substantially changed this. It is strange that a theory should be criticized for failing to explain what it was never intended to explain. This incompleteness in no way invalidates the usefulness and power of evolutionary theory for understanding the millions of organisms that populate our planet today. It simply means that we still need an origin theory that meets scientific standards of experimental verification and compatibility with other established bodies of scientific understanding.

Turning to the "problem" of major innovation, it is increasingly clear that if there ever was a theoretical problem, it was that the mechanisms of organ formation during plant and animal development were too poorly understood for anyone to explicitly envision, step by molecular step, how major morphological innovations could come about by mutation and natural selection. The required events were just too vaguely defined. But this situation is changing rapidly. In the past twenty years enormous progress has been made in unraveling the molecular logic of organ formation. Discoveries are providing plausible detailed scenarios

for how organs might have evolved from ancestors that had no such structures. This general field is often referred to as “EvoDevo” (Evolution of Development) and is currently one of the hottest areas of research in molecular cell biology.<sup>7</sup>

Evolution of life on earth is best understood in terms of the DNA-codes-for-proteins-that-make-organisms-what-they-are paradigm (see chapter 4). Every organism has a characteristic form, physiology, and behavior. Some characteristics are the result of environmental actions such as when wind breaks the limb of a tree or when an animal is small because of poor nutrition, but most obvious aspects of phenotype are inherited from parents.

When analyzed at the molecular level, it is the actions of proteins that determine the phenotype of an organism, and what proteins are present is determined by DNA sequences. To recount a few examples, the protective coating of the T<sub>4</sub> bacteriophage is made entirely of proteins, so it is obvious that structural characteristics such as shape and environmental toughness must be due to the proteins of which it is made. Change the proteins, and the characteristics change. Flowers have certain colors because protein enzymes synthesize pigment molecules in the petals; if there were no enzymes there would be no pigments, and petals would have little color. Change the enzymes, and pigments will be different and flowers will have different colors. Flowers develop petals because regulatory proteins dictate basic “blueprinted” patterns of cell growth during flower development. Other regulatory proteins dictate the timing and location in the plant of the regulatory proteins that determine petal morphology. It is a time-dependent spatial network of regulatory protein expression that ultimately leads to the formation of all heritable plant or animal structures.

DNA in cells of the small weed *Arabidopsis thaliana* codes for 16,000 different proteins (26,000 total including duplications), and most of these proteins are involved in more than one aspect of plant phenotype. A phenotype, of course, refers to an organism’s observable appearance. A bacterium produces several thousand distinct protein types, a mammal more than 100,000. Science is far from amassing a complete catalogue of the specific effects of each protein in even the simplest organisms, but in time it seems likely that giant databases will be created listing the phenotypic effects of all proteins expressed in selected organisms.

Because the structure of every protein is totally determined by the nucleotide sequence of the gene that encodes it, the inherited phenotype of an organism is ultimately determined by the sequence of nucleotides in its DNA. These sequences come from parents and are occasionally modified by mutation. Such changes, selected positively or negatively over time, accumulate information and are responsible for the long-term trends we recognize to be the product of evolution.

Repeating what is learned in every introductory biology class, seven principles characterize the evolution of life:

1. Organisms reproduce.
2. All populations of organisms produce more offspring than there are parents. This is a biological insurance policy that usually gets a population through hard times. It also ensures that in good times populations will expand.
3. Inherited characteristics are determined by the sequence of nucleotides that constitute an organism's DNA (RNA for some viruses).
4. Organisms are composed of cells, and every organism's DNA is physically a part of each cell's structure. Because of this, whenever a cell survives, the information stored in its DNA endures; when it reproduces, its DNA is copied; and when it dies, its DNA-encoded information is lost.
5. Individuals do not evolve; only populations of organisms evolve over time measured in generations.
6. Natural populations are made up of similar "kinds" of organisms, but DNA sequences are not exactly the same within every individual of a population. This means that proteins of the same kind produced in different individuals are subtly different, and the resulting organism characteristics differ subtly. Geneticists call these differences the "heritable variation in a population."
7. Nature is unforgiving. When an individual is in the wrong place at the wrong time, or doesn't get enough to eat, or is unsuccessful at finding a mate, too bad. The dead, impaired, and unlucky don't leave any offspring, and their DNA sequences do not continue on into another generation. All heritable phenotypes that exist today are based on DNA sequences that have "made it," thus far.

An absolute law of nature is that no population can continue to expand for very long. This is no less true for the human population than any

other. The earth is simply not big enough. Bacterial growth vividly illustrates this. Consider a single bacterium that divides, creating two where one was before. In ten generations, if none die there will be 1,024 bacteria, a thousand-fold increase. In twenty generations, the population will be one billion ( $10^9$ ). These numbers are no surprise to a microbiologist who routinely grows billions of bacteria from a single cell by overnight incubation in suitable growth medium. But what happens if we keep going? In 100 generations, if no bacteria were to die, there would be  $10^{30}$  bacteria, and in only 132 generations the mass of bacteria would equal the mass of the earth! Allow fifty-three generations more (185 total), and the mass of bacteria would exceed the mass of the known universe. Now, consider that bacteria produce only two offspring per generation; oysters produce millions. Human beings average eight to ten (per couple) if they do not take steps to limit the number of children they have. If humans today were to begin doubling their numbers every generation (average of four children per couple), humanity would equal the mass of the earth in only 1,175 years (forty-seven generations)! Obviously this cannot happen. Either we control the number of our offspring consciously or nature will do it for us.

A corollary law of nature is that organisms either limit their reproduction or, on average, most offspring must die without reproducing. ***All exceptions are only temporary.*** Since few organisms consciously limit their reproduction, interactions with the environment determine which offspring survive to reproduce and which do not. In nature the choice is made in two ways, either by pure chance or because some organisms are better at surviving and finding mates than others. Organisms interact with their environments in myriads of ways, and the details of these interactions determine whether they eat or starve, or are eaten, and whether they reproduce or not. Given the variations present in natural populations, it is inevitable that some individuals will be better at *something* and some will be worse at *something*. Whenever these *somethings* affect reproduction, they impact the genes present in future populations.

When recounting this story, proteins and genes are always in the background. Genes determine proteins that determine phenotype; so if a heritable trait provides an advantage or disadvantage generation after generation, the corresponding gene sequences will increase or decrease in frequency across the population. When the frequency of a gene

sequence increases in a population, this means that more individuals of a generation carry it and express the corresponding proteins than in the previous generation. The power of even mild selection is impressive. Imagine a particular DNA sequence that provides a 1 percent reproductive advantage or disadvantage, too little for anyone to notice without a careful statistical study. The discipline of population genetics provides methods for calculating the fate of such a DNA sequence over time. What happens over many generations depends on the size of the population and a few other parameters, but if we assume a population of 10,000 individuals in which mate selection is more or less based on chance, then a DNA sequence that provides a 1 percent advantage would spread throughout the population in about 2,000 generations. Depending on the organism, this would take a month for fast growing bacteria or 40,000 years for oak trees and human beings. Just as important, a sequence that imparted a 1 percent disadvantage would take a similar length of time to disappear if it were widespread to begin with, and much less if it were initially rare in the population. A million years provides time for many changes to occur.

Given that natural populations do exhibit heritable variation, some traits will inevitably increase and some will decrease over time. As advantage or disadvantage mostly results from environmental interactions, even small changes in the environment can switch a trait from advantageous to disadvantageous, and back again. Over long periods of time, if the environment is not too variable, the positive and negative selection of traits causes the genetic composition of entire populations to shift; and if certain traits such as brighter colors, or faster running or better disease resistance are favored, then these features will come to characterize the whole population.

The evolution of living things is diagramed in figure 5.3 as a cyclic process. Selection is accomplished by reproductive success, and reproductive success is determined by complex interactions between individuals and their environments. It should be pretty clear that the cycle in figure 5.3 is a special case of the cycle in figure 5.2. Because DNA encodes information, figure 5.3 diagrams a computation. DNA-encoded information is both the input and the output of this cyclical process. Cellular machinery that translates this program into proteins and then into cell and organism structure carries out the computation. One essential function of an organism is to reproduce. Errors in DNA

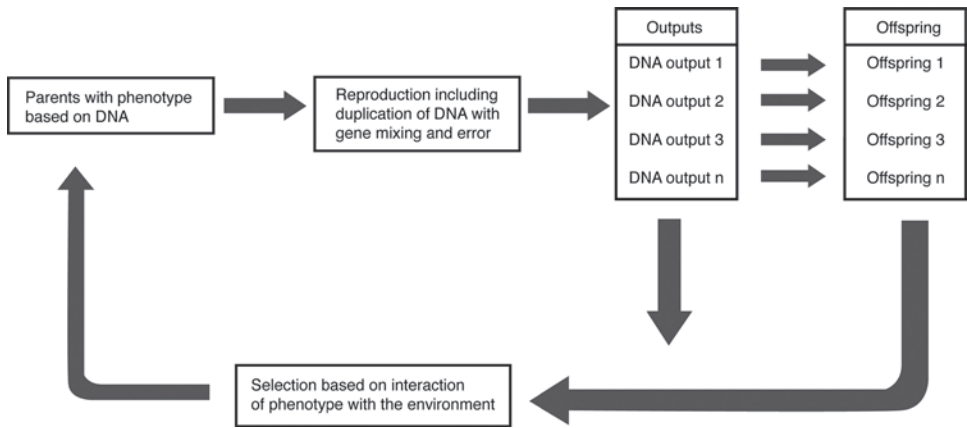


Figure 5.3. The computational cycle of biological evolution.

replication introduce (usually) minor changes during each cycle. On average, more offspring are conceived than there are parents. Because heritable variation in populations is essential if evolutionary change is to occur, various biological mechanisms in addition to simple error have evolved over eons of time to assure an effective supply of new variants and to limit changes from being too big. These include mechanisms for rearranging DNA sequences such as random assortment of chromosomes and recombination that will be discussed later.

Thus, when we focus on the DNA cycle, it is clearly true that the evolution of life is a computation; and because of this, all living things are a product of this computation. DNA sequences participating in this computation accumulate information over time one nucleotide or bit at a time. The resulting accumulated body of information enables cell and body structures as well as behavioral tendencies that are ever more useful (to the organism). These outcomes of the accumulated information interact with the environment. Success is measured by reproductive success. As we will see next, very similar qualities can be incorporated into electronic computer programs.

Figure 5.3 is simply the Darwinian theory of evolution in diagrammatic form. But, this way of visualizing it also makes clear that figure 5.3 is a special case of the computational strategy diagrammed in figure 5.2. Accordingly, Figure 5.2 describes a *generalized theory*



*of evolution.* The engine of complexity is more general than the standard biological theory of evolution because it is not limited to life. It may operate in any system capable of carrying out the computation. This also means that the computational engine of complexity diagramed in figure 5.2 provides a definition of evolution applicable when change occurs in settings that feature populations of entities that are not organisms. As we will see, this definition is quite useful when trying to decide whether computer programs or societies or thoughts evolve in the true Darwinian sense, or whether they change in some other way.

---

## Algorithmic Evolution

### Can computers learn?

Texting on some cell phones can be pretty tedious. To make it easier, phones often include a feature that guesses the word you are typing. The more you use it, the better the phone's ability to guess correctly. It does this by learning your habits. Computer scientists have devised a variety of techniques that allow computers to learn. Many of these employ variations on the engine of complexity. In the world of living beings, evolution results in the accumulation of information about the environment—the system learns. Natural selection when provided small random DNA sequence changes picks those that produce organisms that “fit” better to their environments. Evolved fit reflects information gained by the system about whatever is being fit to. Fish are adapted to life in water and birds for flight because a lot of information relevant to living in these environments is stored in their respective DNAs. One of our principal interests in this chapter is the origin of information that specifies a good fit. We will see that there are really only two possibilities: either information flows from whatever is being

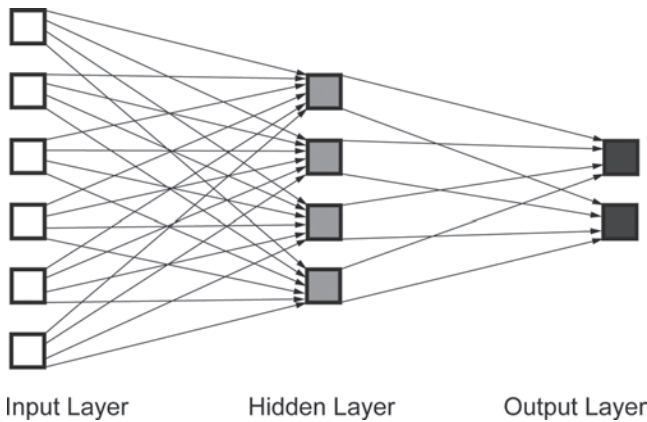
fitted to, or the information is extracted from initially random happenings by selecting those happenings that improve the fit and discarding the rest. In either case it is appropriate to view the resulting information as learned.

Computers have two distinct advantages over human brains: they are fast and they can acquire exact copies of preexisting information. But, learning is more than copying. Learning involves generalization. When you download a program or music file, your computer stores an exact copy of the data structure. Computers do this very well. When you learn a song or a lesson, something quite different happens: you create neuronal connection patterns that represent the song or lesson in ways that evoke certain behaviors and allow you to place the new memory in multiple contexts. You do not recreate the same neuronal pattern that is in the head of your teacher or the composer. It is easy for us to learn general ideas and much harder to remember something exactly. If you don't believe this, just interview three witnesses of an accident or other major event.

The notion of computer learning is easier to recognize than to define. It always involves the acquisition of new information and is more than simple memorization. The term *learning* implies an acquired capability for inference. Something learned allows a computer to solve problems it has never before encountered. This is one of the hallmarks of intelligence. For this reason, computer learning is sometimes called artificial intelligence.

A common learning task is pattern recognition, like recognizing a face in a crowd. A simple scheme for mechanistically learning to recognize patterns is known as an artificial neural network (ANN). In computer science, the term *neural network* refers to a logical system organized as an orderly graph such as the one shown in figure 6.1 and does not refer to a system based on biological neurons.

Neural networks are characterized by three features: nodes, connections, and weights. Each node (square in the figure) is a logical device that accepts inputs, computes a value, and outputs that value on the connections leaving it. Nodes are connected to one another, and each connection has a weight that determines how much effect the connection has on a receiving node; the larger the weight, the more the impact. The network shown in figure 6.1 has three layers, and each node in one layer is connected to all nodes in the next higher layer. An



*Figure 6.1.* An artificial neural network (ANN). Arrows indicate connections between nodes with weights and directions. Nodes are depicted as boxes.

ANN must have at least two layers and may have more than three. It is not necessary for all nodes in a layer to be connected to all nodes in the next layer, and backward connections are sometimes permitted. The ANN in figure 6.1 has only forward connections.

ANNs work in the following way. Input signals (information) in the form of voltages or symbolic zeros and ones are supplied to the input layer. These inputs taken together constitute a data set to be processed. The signal present at each input node is relayed to hidden layer nodes by connections shown in the figure as arrows. The relayed signals are modified by a multiplicative factor called a weight. Weights can be either positive or negative; if they are zero, then no signal is transmitted along that connection. Each hidden layer node processes its weighted inputs according to simple, preestablished rules. For example, it might average its inputs. After processing, each hidden layer node relays a signal to the output nodes. These signals are also weighted, and each output node calculates a value based on its weighted inputs. These final output values constitute the “answer.”

With this scheme, output values are deterministic; they depend entirely on input values and weights. Some ANNs produce complex outputs (many output nodes), and others yield only simple output (one or two output nodes). The ANN in figure 6.1 has two outputs. These could be two numbers or two voltages and could be interpreted

as simply yes or no, or perhaps yes, maybe, and no. If, for example, the inputs characterize individual pixels of a digital photograph, then yes or no outputs might answer the question: Is your favorite aunt in the picture?

If an ANN is to correctly identify your aunt in pictures, it must be trained. Training involves setting the weights so that whenever the input has certain features, a particular output pattern will be generated. The general procedure is to provide the ANN with a variety of inputs, some of which are positive examples and some are not. If the goal is to identify images that include a picture of your aunt, the weights must be adjusted so all pictures containing aunt images yield a yes output and all pictures not containing aunt images yield a no. Surprisingly, it is possible to find such a combination of weights (provided there is something unique about all images showing your aunt). After training the ANN will also be able to identify pictures of your aunt even when presented with pictures it has never before encountered. Various commercial software packages are currently available that use this learning scheme.

There are two distinct strategies for finding suitable weights. One is conceptually very simple: just make small random changes to weights, and use those that give better answers to seed another round of small random weight changes. In other words, evolve suitable weights. This strategy works surprisingly well, but it is slow. The alternative method is to calculate weights by a procedure known as back propagation. Back propagation requires that the desired output be known in quantitative terms (such as a voltage or number) and requires that a difference between actual outputs and desired outputs at each node can be calculated. The sensitivity of a node's output to its various input weights must also be calculated. When such a characterization of each node is possible, new weights can be calculated after each training run so that the new outputs are closer to the ideal. As more training sessions are performed, the performance gets better and better.

Back propagation is faster than random change with selection and is usually preferred when it is possible. When it is not, such as in situations where target numerical output values are not known, or when there is no optimal answer, then a strategy based on random change is the only option. An example of this might be using an ANN to help in the design of a better airplane wing. The best answer is not known,

but appropriately trained neural networks can suggest solutions to be tested when new design constraints are introduced.

ANNs are just one of many strategies that have been devised for computer learning.<sup>1</sup> In general all involve the dividing up of a huge space of possibilities into two or more subspaces so that all positive examples are in one subspace while all negative examples are in the other. For ANNs the configuration space consists of all possible weights and inputs. For even a modest-sized ANN the number of different combinations of possible weights and inputs is too large to try them all. In this situation generalization rules must be developed. All computer learning strategies involve repeated episodes of computer training, and most involve the adjustment of parameters during training rounds by calculating new values that do a better job than previously used values. If ideal values can be computed directly, then training is not required, and learning is not necessary. The answer is simply calculated.

Computer learning algorithms allow a computer to acquire information about a training set. The simplest way for a computer to do this is to memorize all the examples in the entire training set, but this requires a large memory and does little good if a future problem is not included in the training set. To solve unknown problems, information about a class of problems must be stored in a generalized way. The weights in a trained neural net are an example of generalized information that allows neural nets, when properly trained, to correctly characterize patterns never before encountered.

Back propagation is a powerful deterministic technique used by engineers, but is only possible when information can flow back through a mathematical transformation of environmental values into weight values. Many problems cannot be constructed this way. If the needed information is not already known or cannot be extracted from the desired answer, then what other source of information is there? The surprising answer is random guesses with selection. Random guessing will eventually answer any question; the problem is to find a practical way to use this disorganized information source.

In most situations good answers are hopelessly lost in a sea of bad answers when a learning strategy is based on random guessing and nothing else. What should by now be obvious is that a way around this difficulty is to *evolve* an answer. It turns out that many practical problems have the property that there are lots of solutions, some better than

others, and the better ones can be improved upon by making small changes in them. Investment strategies, airplane wings, and transportation grids are just a few examples. Many problems faced by engineers have this property.

A key aspect of small improvements is they are *not* wildly improbable. The engine of complexity takes advantage of the reasonable probabilities characterizing small improvements even though large improvements are hopelessly improbable. The ease of finding good solutions in this manner is made clearer by examining some simple evolutionary algorithms.

### What are evolutionary algorithms?

Not all evolution involves living creatures. As early as the 1950s computer scientists were creating algorithms that employed principles borrowed from biology but that did not involve any of the molecular structures found in cells. In the 1970s and 1980s such algorithms were extensively explored by John Holland and his students at the University of Michigan.<sup>2</sup> Some of these programs exhibit behaviors very reminiscent of biological evolution even though they are nothing but computer software. The existence of such computer algorithms proves the process of evolution is not limited to living systems. They also illustrate the close relationship between evolution, learning, and computation.

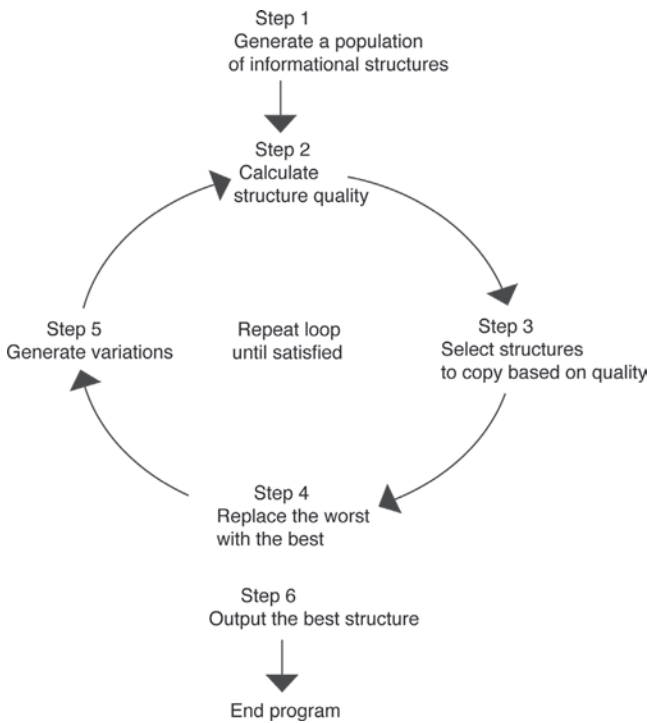
Evolutionary algorithms vary enormously in their details, but always have the following six elements:

1. They maintain multiple copies of information (information bodies) that are potentially useful for something. Often the information represents tentative solutions to some problem.
2. There are “generations” of these information bodies. Each generation includes new copies that are derived from the previous generation.
3. The information bodies can change randomly from generation to generation. As a result, members of the population are not identical.
4. All changes are small in scope and magnitude. Unfettered large random change is useless.
5. During each generation, some information bodies are copied more frequently than others.

6. Decisions determining which information bodies to copy, and how frequently, must be based on consistent criteria and must be based on information encoded by each information body.

An algorithm is a logical list of steps to be followed in a specified order. Evolutionary algorithms have a special feature: they include the particular loop-type organization shown in figure 6.2.

In the first step of this algorithm, initial structures are generated, typically by random means; these can be considered tentative solutions and need not be good ones. Step 2 is assessment. Each informational structure is evaluated against some criterion. Assessment is necessary and nonrandom. Steps 2 through 5 are repeated over and over resulting in the accumulation of useful information culled from random changes



*Figure 6.2.* The fundamental structure of an evolutionary algorithm (adapted from Ashlock 2006).



made. Step 5 is the source of the new information. Because only the worst structures are replaced, the system does not “slip back” even if all the mutated copies are worse than their parents. The overall effect is to accumulate randomly introduced information that improves assessment. If no improvement occurs in a cycle, that is not a problem; there is always another cycle. Long-term change is made by repeatedly building on the best of each previous generation (according to the selection criteria). When a preestablished goal is met or a time limit reached, the program delivers the best result achieved and stops.

This general scheme will produce structures that have higher and higher “quality” (better and better assessment scores) for an incredibly wide range of problems, provided only that variations do arise in the property being assessed. Practically any information-encoding scheme can provide the basis for an evolutionary algorithm. In this chapter we will consider two; many others have been explored by computer scientists. Likewise, the nature of the assessment of quality and the nature of the rules invoked to select which information bodies are to be copied are pretty open ended. Consistency is essential, and the variations generated in Step 5 should not create variants too different from the parental types. If new variants are very different from the previous generation then improvements are rarely if ever generated. At the other extreme, if the variations are too small, then nothing much happens either—each generation is essentially the same as the previous generation.

With the right amount of variation, each generation has a reasonable potential for producing a few information bodies with quality superior to those present in previous generations. When this happens, the information responsible for a better assessment is passed on to the next and future generations. When this strategy is carried out repeatedly, the whole population of information bodies gets better and better at whatever is being selected for.

The scheme presented in figure 6.2 ensures that each generation can never be worse than the previous one by only mutating copies and leaving unchanged the best information bodies. This feature is not always necessary. In many applications, if the mutation rate is not too high, all members of the population can be mutated and the algorithm still works well. To ensure that variations never envisioned by the programmer are tested, at least some aspect of the variations introduced should be random.

A subcategory of evolutionary algorithms are called genetic algorithms because they closely mimic biological evolution. Information is encoded in a linear sequence of symbols; and all members of a population are subject to a low rate of mutation. Mutation is achieved by randomly changing single symbols (for example, randomly changing a 1 to a 0 or a 0 to a 1), and they usually also incorporate a mechanism for swapping subsets of information between different members of the information body population (analogous to sexual reproduction and genetic recombination in biology). Other change mechanisms found in biology, such as duplication of subparts or deletion of subparts may also be employed.

Genetic algorithms have the general property of a mutation rate “sweet spot.” If the change mechanisms are too conservative, change will be very slow, and if mutation is too rapid, the system will collapse into random chaos with little or no memory of past generations being maintained. When the rate of change is properly “tuned,” populations can rapidly evolve in such a way that information carried by each member of the population satisfies the assessment criterion better and better over time. We will now look at two examples illustrating how genetic algorithms work.

#### EXAMPLE 1: ONE-MAX, A VERY SIMPLE PROBLEM.

This computer program transforms a list of 1s and 0s of any order into a sequence of all ones—a trivial problem for a person, but not so easy for a computer if it hasn’t been told how to recognize a “1”. This problem clearly illustrates how genetic algorithms work. Figure 6.3 diagrams a genetic algorithm that accomplishes the task without explicitly telling the computer how to “recognize” a 0 or a 1. For simplicity the sequence in this example is only ten digits long, but it could be of any length.

In this algorithm the loop repeating steps 2, 4 and 5 is the loop characteristic of evolutionary algorithms shown in figure 6.2. The loop that repeats steps 2 and 3 generates ten trial structures for each generation. To give a specific example: let the initial input for the algorithm be 1001000100 (one of 1,024 possible starting sequences of 0s and 1s). Table 6.1 records the progression of a computation starting with this sequence. The final output, as expected, is 1111111111.

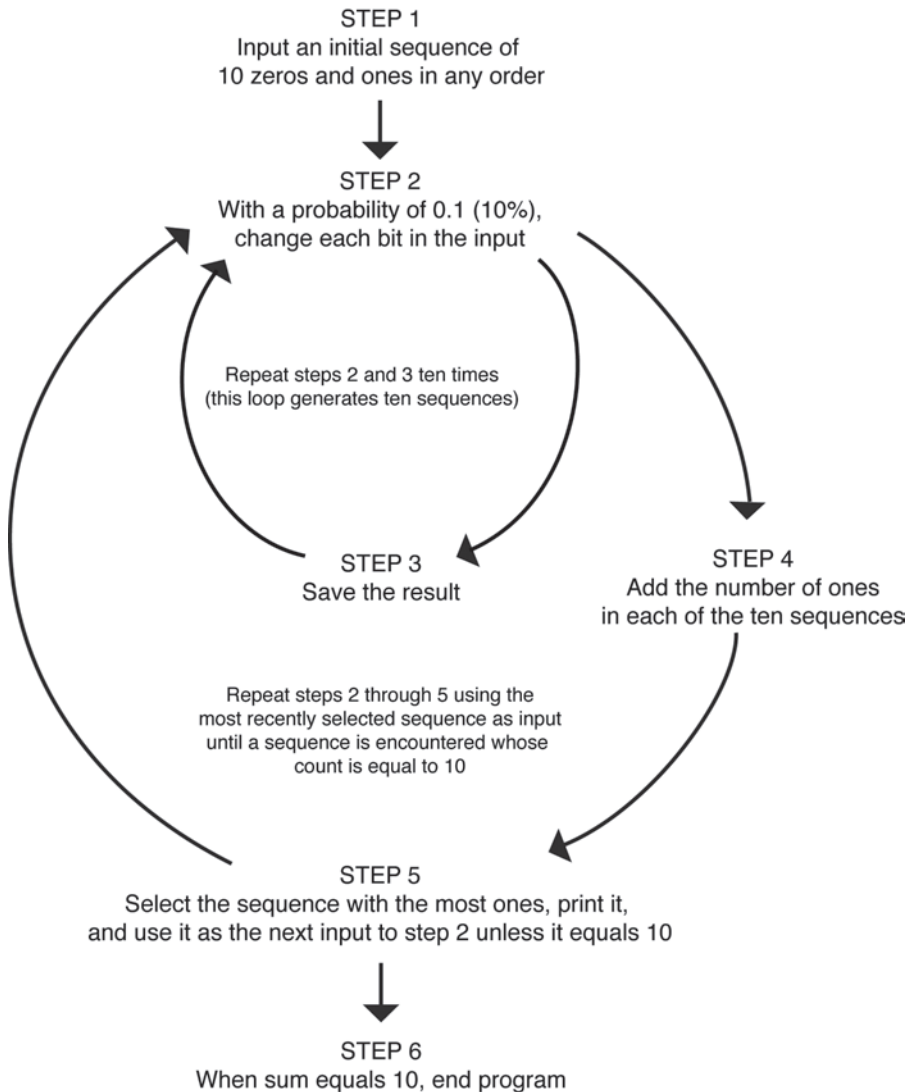


Figure 6.3. An evolutionary algorithm that solves the one-max problem.

For each round of computation (iteration) shown in table 6.1, the ten outputs differ from the input string to that round by 0, 1, or 2 bits. In other words, 0, 1, or 2 bits were chosen at random and flipped (0 turned into a 1 or 1 turned into a 0). This reflects the 10 percent probability of a bit changing each time a string is copied. For convenience,

Table 6.1  
Progression of a computation based on the algorithm given in figure 6.3 and starting with 1001000100.

Iteration 1		Iteration 2		Iteration 3		Iteration 4		Iteration 5	
string	count	string	count	string	count	string	count	string	count
1001100100	4	1011010101	6	1011011101	7	1011011101	7	1101011111	8
0001000100	2	0001010101	4	<b>1111011101</b>	<b>8</b>	0111011101	7	1111011011	8
1001000100	3	1001110100	5	1011010101	6	1111011101	8	1011011111	8
1001100100	4	1001010101	5	1011111001	7	<b>1111011111</b>	<b>9</b>	0111010111	7
<b>1001010101</b>	<b>5</b>	1001000101	4	0011011101	6	1101001101	6	1111011110	8
1011000100	4	1001010101	5	1011101101	7	1110011101	7	0111011101	7
1001000101	4	1000010101	4	1011011111	8	1110111101	8	<b>1111111111</b>	<b>10</b>
1001010110	5	1001010111	6	1011011100	6	1111010101	7	1101011111	8
1001000100	3	1101010101	6	1010011101	6	0111011001	6	1111011111	9
1100000100	3	<b>1011011101</b>	<b>7</b>	1001011001	5	1111011101	8	1110111111	9

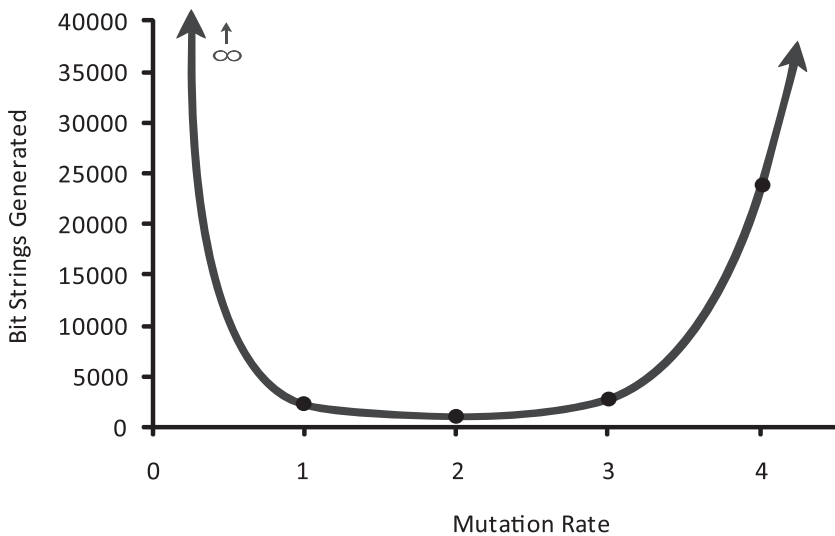
**Table 6.2**

Comparison of the time (number of strings that must be generated) to create a string of all ones by genetic algorithm or by random means.

Number of bits in string	Number of strings needed by genetic algorithm (on average)	Number of strings needed to generate the target randomly (on average)	Speed-up by the evolutionary algorithm
10	60	500	8
50	600	$10^{15}$	1,666,666,666,667 ( $1.7 \times 10^{12}$ )
100	1330	$10^{30}$	$10^{27}$
1000	34000	$10^{300}$	$3 \times 10^{297}$

the one-count is shown in the table for each variant. After each cycle the bit string chosen to serve as input to the next iteration has the highest count and is shown in bold type. For example, all the strings shown in the column labeled iteration 2 were generated from the string having a count of 5 in iteration 1. In this particular experiment, it took five rounds (iterations) to output a string of ten ones. Other trials would not necessarily solve the problem in the same number of rounds. Each time the program is run the details differ because the changes (mutations) made are random, but the final output will always be a sequence consisting of ten ones. The “source” of the ones is random flipping of some zeros to ones. The process also randomly changes 1s to 0s, but these changes are discarded.

For the run shown in table 6.1, a total of fifty strings consisting of ten 1s and 0s were generated. Repeated experiments suggest that the average of many trials is closer to sixty. If, instead of small changes to existing strings, entirely new strings were created during each round, approximately fifty rounds (a total of 500 strings) would be needed to produce one string of ten 1s. Thus, the evolutionary algorithm is roughly ten times faster than a simple random search for a string of ten 1s. For longer strings the time savings are much greater; this is shown in table 6.2. The first row is the example just given.



*Figure 6.4.* Number of strings that need to be generated by the one-max genetic algorithm (figure 6.3) to find a string of 100 ones plotted against the average number of changes made to strings in each generation (mutation rate). At a mutation rate of zero the algorithm takes forever, and at a 100 percent mutation rate, roughly  $10^{30}$  strings must be generated. With a mutation rate of 2 percent, only about 1,300 strings are required.

One thousand bits is not so much information, yet the time it would take to generate a particular string encoding this much information by the random generation of strings is vastly greater than the number of atoms in the known universe multiplied by the age of the universe in nanoseconds (billionths of a second). This means that if every atom in the universe generated a bit every nanosecond, the desired string would never be generated in the lifetime of the universe. From this we must conclude it is impossible to generate a string of 1,000 1s randomly. Yet, it is easy using the one-max algorithm. This illustrates the near magical power of the engine of complexity.

To achieve the speedup reported here, the mutation rate must be optimized for each string length. Optimal performance is usually observed when the mutation rate is approximately  $1/n$  where  $n$  is the length of the input string. The effect of mutation rate for 100-bit strings is shown in figure 6.4.

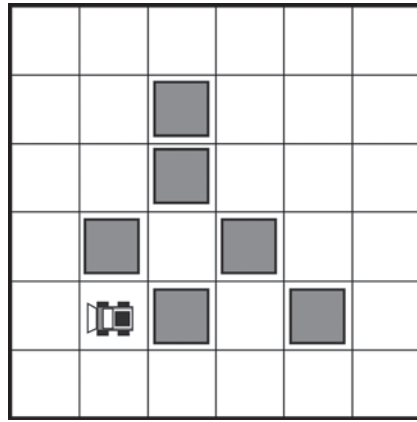
The one-max problem is so simple there is no need for a person wanting to solve it to use an evolutionary algorithm, but the task well illustrates how evolutionary algorithms work and serves as a nice problem for beginning students.

#### EXAMPLE 2: TARTARUS.

Dan Ashlock and his students have explored programs that optimize the behaviors of virtual bulldozers on rectangular boards. The setup is known as Tartarus and was first proposed by Astro Teller in 1993 when he was a graduate student at Carnegie Mellon University.<sup>3</sup> An example is shown in figure 6.5. The task of the bulldozer is to push boxes to the sides and corners of a board. In a simple design, a bulldozer has no sensors and can only push one box at a time. Bulldozers move in discrete steps, and for each step, one of three moves is possible. The bulldozer may move forward one square (F), turn left (L), or turn right (R). A sequential listing of commands, such as FFRFL, is a controller sequence that determines the behavior of the bulldozer. This particular sequence dictates that the bulldozer move two squares forward (FF), then turn right (R), then move one square forward (F), and then turn left (L), in that order. In Tartarus a bulldozer and a box or two boxes cannot simultaneously occupy the same square. If a bulldozer moves forward into a square occupied by a box, the box is “pushed” one square in the direction the bulldozer is moving unless that movement is stopped by another box or a wall. When movement would require a bulldozer to push two or more boxes or a wall, nothing happens, and the algorithm proceeds to the next command in the controller sequence.

Genetic algorithms can create controller sequences that achieve high scores in a limited number of moves (typically eighty moves for a  $6 \times 6$  board). Scoring is one point for each box side that touches a wall. Thus, boxes in corners score two points, boxes against walls but not in a corner score one, and free standing boxes score zero points. Figure 6.6 shows four different outcomes of setups with a six by six board and six boxes. The game can be played on any size board that is at least  $3 \times 3$  and with any number of boxes significantly less than the number of board squares. It is also possible to have more than one bulldozer.

There are two distinct versions of the game. In the simple version, Tartarus is evolved to find a best solution to a particular initial

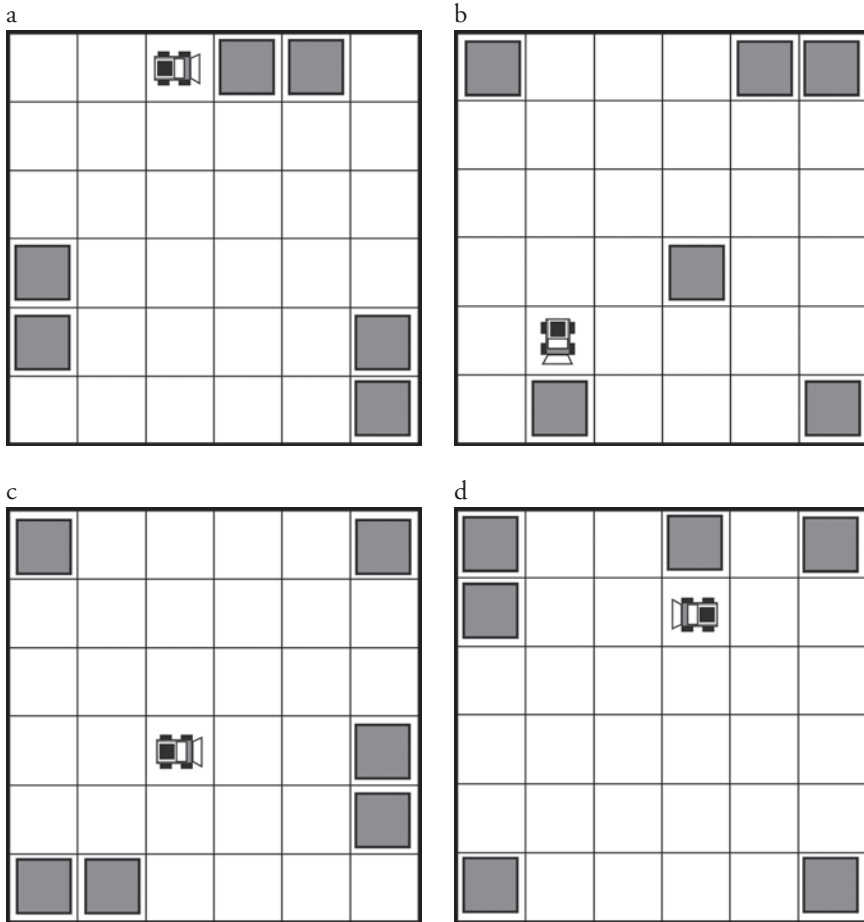


*Figure 6.5.* A  $6 \times 6$  Tartarus board with six boxes and one bulldozer. The figure is modified from one provided by Dan Ashlock (see note 3).

configuration, for example, the one shown in figure 6.5. In the second version, bulldozer behavior is evolved to perform well on a randomly chosen initial configuration. In both versions, multiple controller strings are created, tested against board configurations, and the best selected for mutation and retesting.

When fixed initial configurations are used, one run for each mutation is sufficient to obtain a score for each controller sequence tested; the algorithm, if run long enough, will eventually find the optimal controller. When controllers are tested against randomly configured boards, a number of trials must be run and an average score computed for each controller. After a completed round, controller strings that performed best against a variety of boards serve as the basis for the following generation. Mutation can be as simple as changing a single letter of a controller sequence or may include other change strategies such as selective duplication and/or crossover (also called recombination). In all cases, a population of controller sequences is maintained and controllers with good scores serve as “parents” for future controller generations. Application of the algorithm results in the progressive discovery of better and better controller strings (those that result in higher scores) over time. As with all genetic algorithms, the source of new information and of innovation is (small) random changes made to previously selected controllers.





*Figure 6.6.* Four  $6 \times 6$  Tartarus boards after eighty moves. Panel a: score 7, panel b: score 8; panel c: score 9; and panel d: score 10. The figure is modified from one provided by Dan Ashlock (see note 3).

The game with random board configuration is quite hard, and performance is significantly improved by adding memory and sensors to bulldozers. It turns out to be very useful for bulldozers to remember whether the previous move resulted in a box being moved, or was a push on a box that could not move, or was an attempt to push a wall. Adding memory and sensors makes the algorithm more complicated, but results in much higher scores.

## Can the rules evolve?

In the two previous examples, input to a computer program (a string of 1s and 0s or a controller sequence of letters) is subjected to small random changes in each cycle but the program itself is left unchanged. Genetic *programs* are a subclass of evolutionary algorithms that evolve the program as well as the input. This leads to improved algorithms as well as improved input/outputs.

It turns out that the steps needed to accomplish a task can be represented in a variety of ways. Genetic programs often employ an informational structure known as a parse tree. Parse trees show relationships between operations in a geometrical format. Three distinct ways of representing the solution to a simple task are shown in figure 6.7. In this figure, the task is to add 3 to the product of two variables *a* and *b*. Each of the representations shown is useful in different contexts. Formulas are compact and easy to visualize. Algorithms (in words) are sequential commands that can be carried out by a computer, while parse trees reveal the logical structure of a procedure.

Parse trees are characterized by two kinds of nodes, terminal nodes (or leaves) and operational nodes. In the figure, *a*, *b*, and 3 are terminal nodes while \* (multiplication) and + (addition) are operational nodes. Terminal nodes are not connected to any nodes below. Nodes are arranged in layers and connected to no more than one node above and two nodes below. The size of a parse tree is simply the number of nodes. One and only one node has the special property that it has no nodes above and is connected downward directly or indirectly to all other nodes. This special node is called the *root*. In figure 6.7 the root is the + node.

Parse trees are interpreted from the bottom up. The tree in figure 6.7 contains three layers, and the lowest layer contains the terminal nodes *a* and *b*. These are connected by the operation node \* in the second layer. The structure indicates that *a* is to be multiplied by *b* and the resulting value placed in the \* node. The second layer contains two nodes, the \* node (which now contains the value obtained by multiplying *a* times *b*) and the terminal node 3. These are connected to the operation node + in the top layer. This indicates that 3 is to be added to the value residing in the \* node and that value placed in the + node. Since there

Formula:

$$y = (a * b) + 3$$

Computer Algorithm:

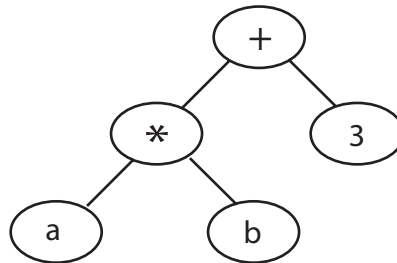
Step 1 enter values for variables a and b

Step 2 multiply the value of a times the value of b

Step 3 add 3 to this product

Step 4 print y = "the value obtained in step 3"

Step 5 end of program

Parse Tree:

*Figure 6.7.* The same logical operation shown as a formula, an algorithm, and a parse tree.

are no nodes higher than +, this is the root node and the operation is completed (the answer is located in the + node).

Every formula and every algorithm can be structured as a parse tree, and parse trees can be evolved by maintaining a population of parse trees, making random changes to nodes and links, and selecting parse trees to serve as “parents” for new parse trees. One effective way for making random changes to parse trees is to pick a node and delete the

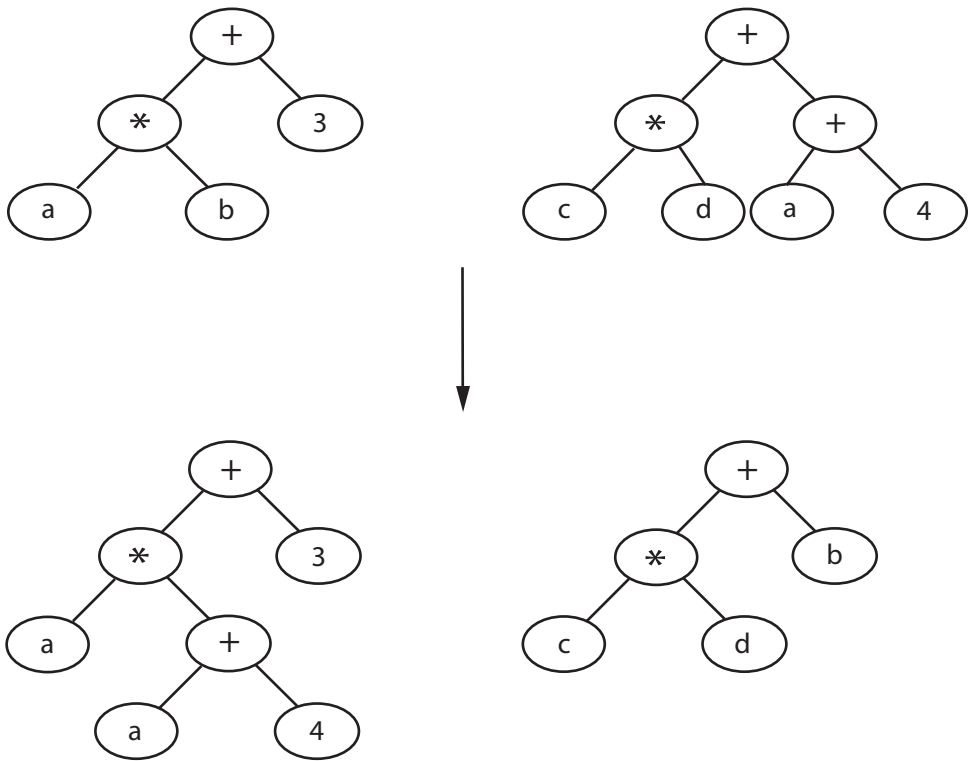


Figure 6.8. Crossover between parse trees. The subtree “a + 4” on the upper right tree is exchanged with the subtree “b” on the upper left tree resulting in the two new trees shown below.

entire subtree that is connected downward from this node, replacing the deleted subtree with a new subtree that has been randomly created from a list of permitted operations and variables. Another way is to swap subtrees between two members of the population. This parse tree version of (biological) crossover is illustrated in figure 6.8.

Parse trees for even rather simple algorithms, such as that in figure 6.3, have many nodes, and are inconvenient to draw. But, with the right kind of problem, and when intelligently employed, genetic programs that evolve parse trees can explore vast spaces of possible algorithms and find creative solutions to some very difficult problems.<sup>4</sup>

I hope it is now clear that evolutionary algorithms including genetic programs are specific implementations of the engine of complexity

(figure 5.2). Each involves populations, mutation, and selection. In the next four chapters we will see a number of other implementations. But, before we get to those, it is helpful to take a look at a traditional way of visualizing any evolutionary process.

### How can we visualize what is happening during an evolutionary computation?

Evolutionary algorithms and genetic programs by their nature find better and better solutions to problems. To better visualize what is happening over time, imagine that all potential solutions to a problem reside in an abstract, many-dimensional space of all possible solutions. Such spaces are usually unimaginably large, but can be imperfectly visualized through a device called an adaptive landscape, which graphically displays the value, or some quality measure, of a large number of potential solutions to a problem in three dimensions. Sewell Wright first used this visual aid in 1932 to illustrate the evolution of biological systems.<sup>5</sup> It works even better for evolutionary algorithms. Figure 6.9 shows the three-dimensional graph of a rather simple mathematical function that for obvious reasons is sometimes called the “hill function.” The graph allows us to see the solution space of this equation by displaying the value of  $z$  for various values of  $x$  and  $y$ .

No matter what values of  $x$  and  $y$  are plugged in, the largest possible value of  $z$  is 1. This can be demonstrated either by using the techniques of calculus or by employing an evolutionary algorithm. Evolutionary algorithms do not know how to solve calculus problems, but can readily find the maximum value of this equation. The algorithm works by creating a “population” of guesses, evaluating each of them, selecting the best (largest value), and then creating a new population of potential answers by making new guesses that are “close to” the best previous guess. To illustrate, we will follow the output of one run of an algorithm that maintains a population of twenty guesses. Each guess is a pair of values for  $x$  and  $y$ . For simplicity only five of the twenty guesses that were generated in each round are shown, and the numbers are restricted to two significant figures (two digits after the first nonzero number). Table 6.3 shows the progress of this run. Only rounds 1, 3, 4, 5, and 7 are shown. In each round the best guess (the highest value of  $z$ )

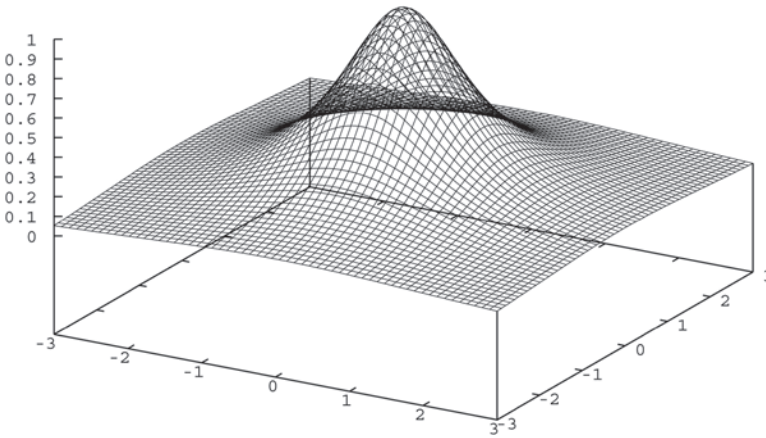


Figure 6.9. A three-dimensional graph of the “hill function,”  $z = 1/(x^2 + y^2 + 1)$ . The figure was kindly provided by Dan Ashlock.

is shown in bold type. Initial guesses were between 1000 and  $-1000$  and subsequent guesses were restricted to an interval four times the size of the best guess. Thus, if the best guess for  $y$  in one round was 3, guesses for  $y$  in the next round were randomly chosen to fall between  $-3$  and  $+9$ . The larger the value of  $z$ , the better the answer. Note that for each round the value of  $z$  is closer to 1.0 than in the previous round. In round 8 (not shown) of this particular run, the value of  $z$  reached 0.998, which rounds to 1.0. To reach the true value of 1 with no rounding took over 100 rounds.

Looking again at figure 6.9, one can imagine the initial guesses scattered across a landscape featuring a region (the hill) where the best solution resides at the top. In the initial round of guesses, none are particularly close to the hilltop, but one is closer than any other. The next round, guesses are restricted to the general neighborhood of the first guess. Once again, one of these is closer to the ideal solution than is any other. In only five rounds, an “answer” of 0.94 is found, and in seven rounds, 0.990. The algorithm is slow to find the true answer of 1.00, but it will get there eventually.

For evolutionary algorithms of all kinds, it is important for each round of guesses to be restricted to the neighborhood (in solution space) of the last best guess; if this is not done there is little opportunity

Table 6.3

Selected values generated by an evolutionary algorithm that finds the highest value for the hill function.

Round 1	Guess 1	Guess 2	Guess 3	Guess 4	Guess 5
X	180	<b>100</b>	940	-60	680
Y	60	<b>-20</b>	-320	740	-700
Z	0.000028	<b>0.000096</b>	0.0000010	0.0000018	0.0000050
<i>Round 3</i>					
X	17	-21	59	<b>6.7</b>	9.6
Y	-42	-8.4	25	<b>8.4</b>	-34
Z	0.0048	0.0019	0.00024	<b>0.0086</b>	0.00079
<i>Round 4</i>					
X	12	<b>3.0</b>	-3.7	13	-0.49
Y	17	<b>2.5</b>	7.2	-3.0	15
Z	0.0022	<b>0.061</b>	0.015	0.0052	0.0.0042
<i>Round 5</i>					
X	8.8	0.72	-1.9	3.4	<b>0.24</b>
Y	4.6	-2.2	5.0	-1.5	<b>0.10</b>
Z	0.010	0.16	0.034	0.069	<b>0.94</b>
<i>Round 7</i>					
X	0.60	1.0	<b>-0.069</b>	0.24	-0.038
Y	0.070	0.37	<b>-0.049</b>	-0.060	0.19
Z	0.98	0.94	<b>0.990</b>	0.94	0.96

for steady gains to be made. Without restricting guesses to the local neighborhood of the last best guess, the algorithm is reduced to making purely random guesses each round, and finding a good solution by random guessing is essentially impossible for most problems because of the large size of most solution spaces. For the hill function example, finding the best answer by guessing randomly would require millions of random guesses, whereas the evolutionary algorithm will find it in less than 200 rounds, and as shown in the example, the algorithm found a pretty good answer (0.94) in only five rounds. A feature of evolutionary algorithms is that they typically find “pretty good” answers quickly, even if finding the best answer takes much longer.

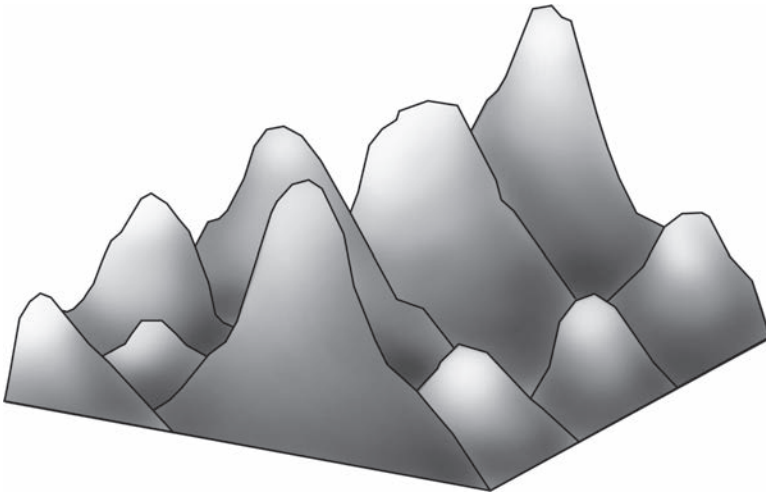
A property of the hill function shared by many problems is that most potential solutions are far from the peak, but when one examines the local region of any particular solution, almost half are closer to the peak than the current working solution. Thus, by restricting each round to the local region of the last guess some new guesses will almost always be an improvement, and the overall search easily progresses to better and better solutions. This contrasts with unrestricted searches where the vast majority of guesses are no better than the one at hand. Global searches only reliably succeed if you make vast numbers of guesses or the search space is small. For an evolutionary algorithm to work well, there should be at least one solution in each round (or in most rounds) that is higher up the hill. By working steadily toward the correct answer, round after round, the best solution at the very top will eventually be found.

The hill function is very easy for genetic algorithms to solve because of its smooth shape. Many problems are not so simple. Solution spaces frequently have many hills and valleys. A hypothetical example is shown in figure 6.10. In this situation it is very easy for an evolutionary algorithm to get stuck on a lesser peak without ever finding the highest peak. As stated before, nearly all solution spaces are characterized by many dimensions. Figures 6.9 and 6.10 illustrate two-dimensional solution spaces (the x-y plane). The one-max problem with ten 0s and 1s in the solution string has a ten-dimensional solution space. Maximizing a 1,000 bit sequence requires search in a 1,000-dimensional space. Spaces with more than three dimensions are impossible to draw, so figures such as 6.10 give one a useful but impoverished image of what higher-dimensional solution spaces must be like.

### Is randomness necessary for learning?

There are four lessons that I hope readers take away from this chapter. The first is that evolution can occur outside the domain of biology. The second is that solution spaces (configuration spaces) are often so large that purely random searches of them for answers are bound to fail. The third is that one can often evolve a good answer even though one cannot be found by random guessing. The fourth is that when one needs new information but has no idea where to find it, randomness is





*Figure 6.10.* A hypothetical two-dimensional adaptive landscape with many peaks and valleys. Each peak represents clusters of solutions that are better than average, while each valley represents poor solutions. Finding a peak is easy while finding the highest peak is often difficult for evolutionary algorithms.

a surprisingly useful source, often the only source, but even then only if the search is managed properly.

The computational view of things makes clear that actions may be deterministic or probabilistic or a combination of the two. The outcome of a deterministic process can be known from the start, whereas the outcome of a probabilistic process is unknowable in detail until it is actually produced. The word *probabilistic* means that randomness plays a role in the outcome. An outcome that is 99 percent certain is probabilistic because it is less than 100 percent certain.

That evolutionary algorithms, and more generally computer learning, work in a wide variety of situations suggests that solution spaces tend to be reasonably orderly, with good answers clustered together and with the best answers being nearby to less good, but still useful answers. This allows learning algorithms to partition multidimensional solution spaces into simply defined regions that separate good answers from poor and allow evolutionary algorithms to successfully “climb hills” and find successively better answers.

Ultimately, if you know absolutely nothing about a solution, random guessing is the only option. Most frequently we know something about the answer, and a modest number of random guesses within the framework of what we know will usually turn up something that is slightly better than the rest; if we then randomly sample “nearby” answers (in solution space), even better ones will be found. Repeating this process typically leads to pretty good answers fairly quickly. If one wants the best possible answer, it may take longer, but even that can often be found.

By making restricted random guesses (sampling nearby possibility space) the odds of finding a solution better than any currently in hand are surprisingly good. By simply following a trail of better and better solutions, one can find a very good solution quickly and easily. This is the “trick” of the engine of complexity. It is what is not understood by critics of biological evolution who say that something cannot possibly have evolved because it is too improbable. All learning algorithms take advantage of an underlying orderliness in our natural world that allows the strategy to work, and the engine of complexity is a learning algorithm. Mathematicians can construct solution spaces that are not so orderly and on which evolutionary algorithms don’t work, but fortunately our universe is not built that way.

Ultimately, all successful learning algorithms acquire information. Such algorithms, of which the engine of complexity is one, do not require access to randomness, but without such input, the learning process is deterministic; what may be learned is either built into the algorithm or is imported from another source. Appropriately restricted random sampling of solution spaces paired with selection turns out to be a remarkably powerful way of acquiring information. Focusing on the engine of complexity, it is appropriate to say that *the algorithm functions by extracting useful information from randomness.*

# 7

---

## Evolution Within the Body

How does the human body form from a single cell?

Picture yourself in Paris. Most tourists to the City of Light spend a day or more at the Louvre. In the Greek and Roman antiquities section of the museum around the corner from the more famous Venus de Milo, at the head of the Daru staircase is a magnificent sculpture called the Winged Victory of Samothrace. The figure, even missing head and arms, is stunning. How did the sculptor do it? I am told that great artists see clearly in their heads what the final product will look like. The mental image serves as a blueprint, but how does that blueprint get converted into a sculpture? The detailed procedure differs somewhat depending on the medium. A clay sculptor has several options, while a stone sculptor simply chips away. For both, the final sculpture is the outcome of a tedious process in which material is added (clay) or removed (both clay and marble). Every addition or subtraction establishes a form that more nearly approximates the final goal. Each intermediate state also serves as the basis for the next addition or removal.

Nature makes things in a similar fashion, but without a blueprint. Snowflakes are the outcome of water molecules adding preferentially to those parts of the growing structure that are most exposed to air. Every water molecule that attaches alters the target for subsequent molecules. By following a few simple rules, intricate shapes get constructed. A craggy mountain is formed like a stone sculpture. Land is uplifted, and rain and ice scour away parts of the landscape leaving peaks and valleys. Sculptures of nature are not predetermined, but the processes by which they come about have many similarities to the creation of three-dimensional art.

Sculpting is a low-tech activity. Primitive peoples everywhere carve figurines out of wood and stone. The ancient Greeks who produced the Winged Victory, the Venus de Milo, and thousands of other magnificent sculptures were masters of a simple technology. They didn't have computer programs allowing detailed three-dimensional predesign of their sculptures or that would direct industrial robots to reproduce their exact designs by laser ablation of marble blocks. The Winged Victory may have been preceded by a sketch, but detailed instructions like those used to create a screwdriver in chapter 4 or that are created today by engineering CAD/CAM (computer aided design/computer aided manufacturing) programs were never used. In chapter 5, I introduced the engine of complexity: a computational procedure that assembles information needed to specify purposeful structures. Does a sculptor somehow employ this same strategy? I argue that something similar does underlie sculpting, but it is not exactly the engine of complexity. The element missing from the cycle shown in figure 5.2 is multiple repetitive copying.

Cumulative structure formation is observed everywhere in nature. Most natural things are built by repeated addition to and subtraction from already existing structure. This logic is not so different from the engine of complexity, but it lacks any requirement for multiple copies. Earlier, I briefly introduced the idea of computational chains. These take the general form: body of information 1  $\rightarrow$  manipulation  $\rightarrow$  body of information 2  $\rightarrow$  manipulation  $\rightarrow$  body of information 3, and so forth. Copying plays no essential role in such chains; each parental information body provides a direct basis for its progeny information body. When a snowflake forms, "manipulation" is a water molecule adding to the growing structure. Information is inherent in all physical

structure because information and structure are just different ways of looking at the same thing. So, from the information perspective, all physical processes are computational chains. When a chain has the form of a repetitious loop, it has many elements of the engine of complexity but may not include the formation of multiple copies. To help us keep things straight, let's call cumulative loops that lack copying "engines of construction." The Winged Victory and snowflakes are both examples of what can be achieved by the engine of construction.

Stone sculpting does employ selection. When considering where to place her chisel, the sculptor entertains multiple possibilities before each swing of the mallet. Selection of the next spot requires comparison of the current structure with a mental image of the desired final structure. Water molecules added to a growing snowflake are also selected—many water molecules approach a growing snowflake for every one that binds—but this selection is not based on comparison to a mental image or any other kind of blueprint. The rules followed are simple, straightforward, and statistical.

To make organized sense of all the different ways that things get made, we can place the physical formation of objects in a hierarchy of computational processes shown in figure 7.1. With the possible exception of the origin of the universe, all structure formation begins with preexisting structure of some sort, so essentially everything is included in the top level. The first partitioning separates deterministic from probabilistic processes. Probabilistic processes involve random actions of some sort. Anything involving thermal molecular motion falls into this division. On the deterministic side are activities that do not permit any random actions. In this category are most electronic computations and the operation of many mechanical devices, such as windup clocks. The probabilistic side is further partitioned into sequential and iterated. An iterated process involves the same actions repeated over and over, while sequential processes are characterized by more varied actions, one following another. Most processes in the physical world are sequential, though some such as crystal formation, are iterated. The iteration branch is further divided into phenomena that involve copying and those that do not. The lowest level of each branch of the hierarchy distinguishes between structure based on simple rules and structure requiring complex rules (i.e., instructions). The somewhat arbitrary division between simple and complex rules characterizes every terminal

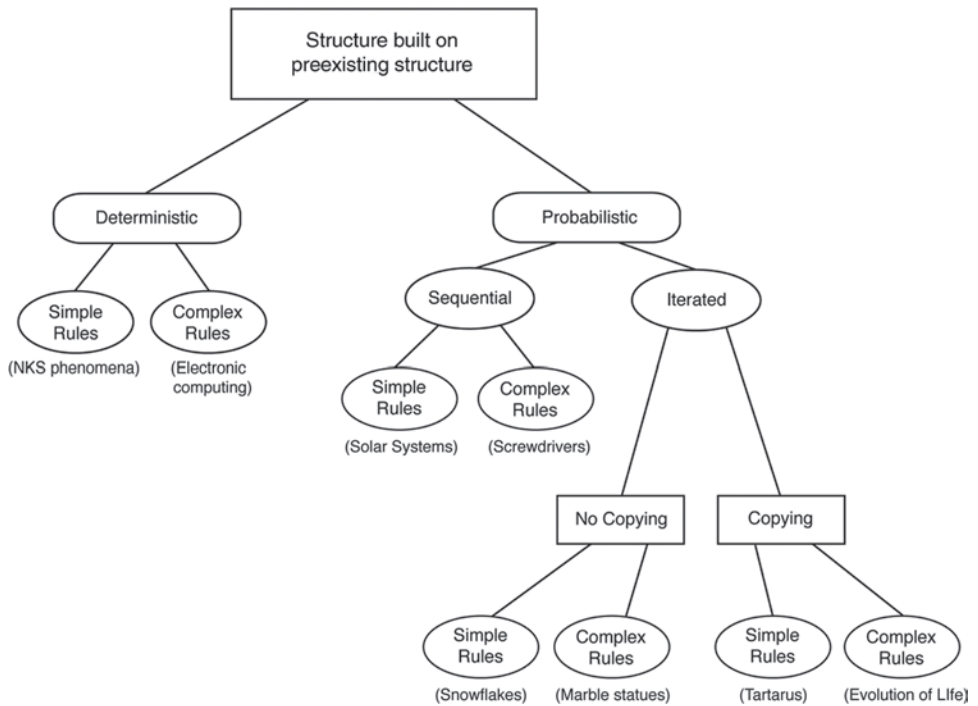


Figure 7.1. The structure hierarchy. An example of each terminal branch is shown in parentheses.

level of the hierarchy. The deterministic side can also be divided into sequential and iterated, but such a division leads to no further insight, so it is omitted.

The engine of complexity is located in the lower right-hand corner of the figure; this is where repeated probabilistic phenomena involving repetitive copying are located. The engine of construction is in the probabilistic iterated no copying branch. Snowflakes and marble statues fall here.

The engine of construction is capable of generating very complex structures. Perhaps the most impressive example of what can be accomplished by repeated selection in the absence of copy formation is the creation of a human body from a fertilized egg cell. Creation of a body is very much like sculpting, but since there is no sculptor, it is also a bit like snowflake formation. Human beings are incredibly

complex. Adults are made up of an estimated 10 trillion ( $10^{13}$ ) cells. There are over 200 distinct cells types and uncounted minor variations. The cells are organized into tissues, organs, and organ systems that taken together exhibit far more purposeful detail than can possibly be encoded directly in the DNA of a fertilized egg. Everyone starts as a single cell, and whatever DNA-encoded instructions there are for the new human being are stored in that cell. So, scientists and philosophers alike would like to know how the body develops from a simple cell even though there is not enough DNA to encode detailed instructions telling each cell what to become and where to go.

We have a rough idea. Science tells us that cells in the body divide, change their protein expression patterns, and die in response to environmental signals. For the most part other cells provide the environment. DNA provides a blueprint, of a sort, or perhaps we should say a sketch, and structure is created by adding cells here and removing cells there, incrementally building toward the final form. Viewed broadly, body formation is the outcome of excessive cell production followed by elimination of those cells that do not satisfy local criteria. In this process each cell follows simple local rules that are determined by cell history and location. The role of DNA is to serve as a repository for the rules.

Let us consider the human hand. The adult hand is made up of perhaps 100 billion ( $10^{11}$ ) cells, 1 percent of the body. These cells are organized into skin, muscles, bones, ligaments, nerves, blood vessels, hair follicles, and nail roots (the tissue that grows fingernails). Each tissue includes several distinct cell types and is organized into highly specific shapes. The hand is distinctive with five separated fingers each stiffened by bone and made flexible by joints. It is capable of movement by means of muscle contractions. Nerves connecting to the spinal column and brain coordinate muscle movements. All this structure is surrounded by skin, and all parts are served by blood vessels. Because human hands are pretty similar, and clearly different from dog paws or fish fins, we know the design is more or less specified by each person's (human) DNA.

The hand appears to be much more complicated than a screwdriver, and as we saw in chapter 4, extensive instructions are needed for the formation of a screwdriver. Human DNA also encodes needed information for many other body parts including legs, livers, brains, and

ears, so only a small portion of human DNA-encoded information can be devoted exclusively to hand formation. Much efficiency is undoubtedly gained by using the same genes to specify muscles, bone, blood vessels, and skin wherever they occur in the body. But even so, there must be a significant amount of information devoted to making a hand different from a foot or a nose. The details of how this information is used are not fully known, but some things are clear. First, the hand is not formed directly in its final form, but is created sequentially starting from a largely undifferentiated mass of cells.

Much of the information needed to make a hand specifies location-determined cell fate. The neighborhood is all important for determining what happens with embryonic cells. Two processes play central roles. These are cell differentiation, wherein cells divide and change character over time as various genes are turned on and off, and coordinated cell movements placing groups of cells in new environments. Except for the earliest stages, animal development is characterized by cell migrations that bring cells into contact with new and different neighbors. These migrations are only loosely specified, and cells frequently “get lost.” Generally, the lost cells have one of two fates: they belatedly find their way to their proper location, or they die.<sup>1</sup> Many cells die during development because they are in a place at a time where the molecular signals they receive from surrounding cells do not match their preprogrammed needs. These cells literally commit suicide.

This strategy seems wasteful, but from an information conserving perspective it is efficient. The alternatives would be for the body to tolerate many cells in the wrong places or for every cell to keep a time-indexed database of all possible locations in the body. With such a database a lost cell could, in principle, plot a course from where it is to where it should be at any particular time, or perhaps it could reform itself into a new location-appropriate cell type. As far as we know, this type of information is not encoded in DNA, and it would take a lot of DNA to encode of all the needed information.

A classic example of position-based cell selection is finger formation. Early on in body development, the embryonic hand is shaped like a paddle. Within the paddle, an alternating pattern of gene activity is established that results in the formation of nine alternating zones of cells having different gene expression patterns. The pattern in the first zone stimulates cell division and differentiation leading to



the formation of connective tissue, while the pattern in the adjacent zone causes those cells to commit suicide. Cells in adjacent regions alternately become bone or die. There are five bone-forming regions separated by four zones of cell death. The result is five separated digits (embryonic fingers) where a simple paddle existed before. Future skin cells migrate around the bone-forming regions, creating separate fingers. In informational terms, “pruning away” and “putting on a glove,” is a very efficient way to create this complex shape.

Animal development is replete with examples of selective cell death playing a critical role in the formation of complex structures. Selective pruning is a central feature of both development and evolution, but alone does not mean bodies *evolve* their final forms as they grow. The notion of evolution is more than just survival of the chosen. The central essence of the engine of complexity is copying with minor change followed by selection based on which of the copies best satisfies certain criteria. Selection during development is generally not due to genetic variation but rather to a mismatch between cell properties and environmental signals. Cells in different microenvironments are either stimulated or repressed depending on the environment they find themselves in. Genetic variation among cells plays no role, nor does the copying of structures that have been selected.

To illustrate the similarities and differences between the engine of construction (no copying) and the engine of complexity (copying), we will spend the remainder of the chapter contrasting brain formation with the immune response. Both body systems are characterized by far too much complexity to be directly specified by DNA in cells; and for both systems, every person exhibits a unique pattern of cell structure and molecular level interactions. In both of these body systems cumulative random choice is responsible for the uniqueness we observe, but randomness is generated and used differently in the two systems, and only the immune system couples multiple copy formation with selection.

## How is the brain wired?

The human brain contains more than 100 billion ( $10^{11}$ ) neurons, similar to the number of cells in a hand; but each brain cell averages 10,000

( $10^4$ ) cell-to-cell connections, far more than hand cells. This means the brain is characterized by roughly 1 quadrillion ( $10^{15}$ ) cell-to-cell connections. The connection patterns determine who we are, what we do, what we think, and what we remember. Human DNA has about 3 billion ( $3 \times 10^9$ ) nucleotides. Current estimates of the number of genes based on DNA sequence data are between 20,000 and 30,000 ( $2 \times 10^4$  and  $3 \times 10^4$ ). Thus, if every human gene were devoted to the formation of some specific aspect of brain structure—something that is certainly not true—each gene would need to specify on the order of 100 billion ( $10^{11}$ ) cell-to-cell connections. Alternatively, if every nucleotide in your DNA specified neuronal connections, then each nucleotide would need to specify 300,000 neuron-to-neuron connections. The results of these calculations are obviously silly. We are safe to conclude that neuron-to-neuron connections are not specified directly by genes. Yet, every person has a human brain and not a dog brain or an insect brain precisely because we have human DNA. This presents a conceptual dilemma.

Several features of the brain must be accounted for if we are to understand how immense functional complexity comes about in the absence of exact specification. First, although all normal human brains are grossly similar in the sense that all have the same major anatomical features such as a cerebral cortex, hypothalamus, and so forth, and very similar nerve tracts connecting major anatomical features, all human brains differ enormously in their microanatomical detail. The details of what neighboring cells any particular neuron connects to are different from individual to individual. This is true even for identical twins. It is as if the information in DNA specifies the broad outline or general rules to be followed during brain development, but not the minute details. A major challenge facing modern neuroscience is to understand how brain design enables reliable, complex, and purposeful function without a detailed specification of the connections. The design strategy is utterly different from that used to build digital computers, where every production unit of a particular model is exactly the same in microscopic detail. Without this exacting degree of specification, electronic computers would not work. How is it possible to create an information-processing structure as complicated as the human brain by only loosely specifying the connections? Selection is almost certainly the answer.

Gerald Edelman has argued for more than thirty years that the evolutionary paradigm in a form he calls “neuronal group selection”

is essential for understanding the brain.<sup>2</sup> Evolutionary principles, he argues, are utilized during primary brain formation and also play a fundamental role in learning. We will see in chapter 10 that the act of learning creates another philosophical dilemma when analyzed in terms of information theory and that the engine of complexity provides a potential solution.

The present section is not about learning but rather the formation of structures including primary connections between neurons that are needed for learning to occur—in short, how the brain gets wired. We will see that the pruning away of undesirable or unnecessary connections plays a crucial role. Formation of large-scale brain features presents exactly the same conceptual difficulties as the formation of other body structures such as the liver or a hand, but the specific details of which neurons connect to which other neurons adds another level of complexity. After all, it doesn't really matter which liver cell is next to which other liver cell, but it does matter which brain cell is connected to which other brain cell.

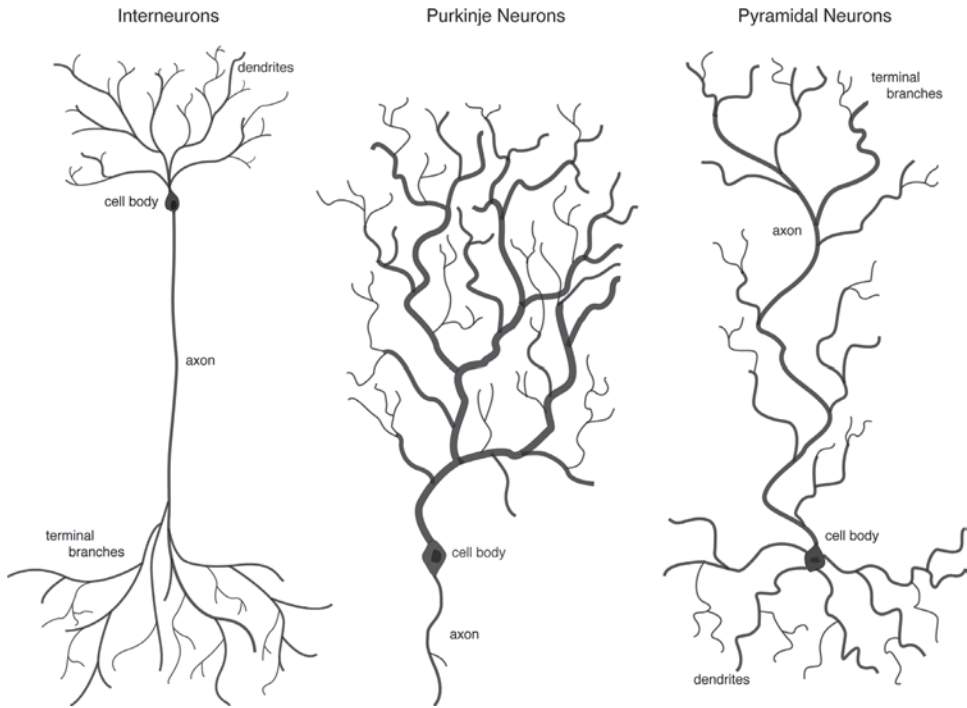
Once positioned in generally appropriate environments by programmed cell migration and the elimination of misplaced cells, embryonic neurons stop migrating and begin extending long, very thin cellular processes that are attracted by long-range signals. As these cellular extensions grow, they explore the local environments they pass through. When an appropriate region is reached, connections called synapses form between cells. When inappropriate cells are encountered, the exploratory processes withdraw and are redirected elsewhere. Not all synapses are functional. When a formed synapse functions, processes within them are strengthened; when a formed synapse does not function, it withers. When a neuron fails to form any functional synapses, it dies. Long-range attractive and repulsive signals channel the growth of neurons, focusing detailed exploration on specific populations of distant cells. In this way, nerves (bundles of cell extensions) project to different parts of the brain and body and make functional connections with distant ganglia, with muscles, or with other targets. This self-wiring activity, though guided by molecular signals, is noisy with possible connections being missed and inappropriate connections being attempted. A central aspect of the process is the strengthening of appropriate connections and elimination of inappropriate ones. Connections are selected according to simple rules that are manifestations

of the structures of particular proteins expressed by each cell, and, of course, those protein structures depend on complicated rules written in the language of DNA.

The cellular processes that extend out from neuron cell bodies and make connections with other cells are called dendrites and axons depending on whether they collect information from other cells (dendrites) or send information to other cells (axons). A few examples are shown in figure 7.2. For obvious reasons the tree-like elaborations of dendrites are sometimes referred to as arbors. In the figure fine branches connect with other cells that are not shown. Synapses occur at the tips and along the sides of dendrites and at the tips of axons. The formation of these structures is geometrically and temporally constrained but not predefined in detail. Neurons of a particular type in a particular region of the brain tend to have similar shapes, but the fine detail is different for each cell. Similarities include the approximate number of synapses, the rough shape of the arbors, and the types of cells connected to. Decisions about which cells and where on those cells connections are made seem to exhibit a significant degree of randomness. As growing processes reach their approximate destinations, they branch and rebranch exploring the local region; some branches make successful connections while others fail and retract. The final arbor shape is determined by a combination of broad genetic guidelines, random local exploration, and environmental detail. Because detail at the cellular level differs from person to person, and because random exploration is, well, random, the final results are always somewhat different from person to person.

During the neuron-to-neuron wiring stage of development, historical contingency is all important. When a branch forms, if stimulated it becomes stronger; if there is no stimulation, it disappears. An obvious feature of this system is that a branch that is no longer present cannot serve as the basis for future branches. In this way, arbor shape encodes a kind of three-dimensional memory of past branching decisions. For each neuron, a unique structure is built up through a process of random branch initiation, semirandom environmental exploration, and selection based on what a branch encounters.

Very similar principles apply to the growth of woody plants. Genetically identical elm trees have the same general shape, but the exact branching pattern differs from tree to tree. Random events at the



*Figure 7.2.* Neurons are characterized by a cell body, an axon, and multiple dendrites. Both axons and dendrites are thin branching extensions of cells that make connections (synapses) with other cells. Dendrites collect information from other cells and axons transmit information to other cells. The exact branching pattern of all but the very simplest neurons is unique but similar for neurons of a particular type.

molecular and cellular levels are selected by environmental interactions, and the final result is a unique three-dimensional structure. In the brain neuronal branches do not extend into air but into a three-dimensional milieu of other cells. Connection details are determined by interactions of the branches with molecular signals encountered and synaptic activity that may or may not be established. Branch growth is stimulated by internal and external signals, but aspects of the branching process are significantly random, making subsequent selection responsible for the creation of nonrandom connection patterns.

To summarize this section, brain development relies on directed but microscopically random exploration combined with selective pruning

of inappropriate connections. Neuronal branches are fairly similar, so they might be considered to be “copies,” but neuronal branches are not uniquely specified in a way that allows their particular details to be replicated in other locations. Structures must always be made anew without the benefit of past experience. Thus, brain development does not utilize a central feature of the engine of complexity, and as impressive as brain development is, it does not tap into the most powerful strategy offered by repetitive selection. A very different body system that does pair imperfect copying with selection is the immune system.

### Why isn't the body destroyed by microorganisms?

Infectious diseases are caused by microorganisms, mostly viruses or bacteria, which can often reproduce in a matter of hours within the body. Because evolution operates on a time scale measured by generations, microorganisms can evolve tens of thousands of times faster than we can. Any specific molecular defense that we might evolve over millions of years can be outmaneuvered by a counter-evolutionary strategy in a few months or years. How is it then we are not constantly being overwhelmed by an avalanche of microbial infection? The short answer is that within each of our bodies our immune systems evolve a molecular response specific to each invasion on a time scale as fast or faster than is possible even for most microorganisms.<sup>3</sup>

Defense against these invaders relies on the identification of molecules (proteins, polysaccharides, and lipids) that are not normally present in the body. The defense system is traditionally divided into two parts called innate immunity and adaptive immunity. Innate immunity features various mechanisms that identify and eliminate most invaders based on their general properties. Part of this system consists of amoeba-like cells that patrol the body looking for and engulfing things that shouldn't be there. Some potential invaders, however, have evolved ways of concealing features that the innate system recognizes. These more challenging invaders are handled by the adaptive system.

The adaptive immune system relies on the identification of uniquely identifying molecules; all organisms have them. This system has two branches: one, called humoral immunity, is populated by what are known as B-cells; and the other, called cellular immunity, is populated

by T-cells. Cells of both branches produce specialized proteins that bind with high specificity to other molecules. B-cells produce proteins known as antibodies, while T-cells produce a class of proteins known as T-cell receptors (TCRs). Antibodies bind to and mark foreign macromolecules (proteins or polysaccharides) present in bodily fluids, while TCRs bind to fragments of foreign proteins that are “presented” by cells that make up the body. Both branches of the system “remember” molecules previously identified as foreign and respond much more quickly to a subsequent invasion by any organism exhibiting the same molecular signature. This memory is what we commonly call immunity.

Both B-cells and T-cells respond by specifically recognizing molecular-scale features made of proteins or polysaccharides called antigens. Since your body also produces proteins and polysaccharides, it is essential that the system recognize foreign antigens, but not self-antigens. At the molecular level, “recognition” means binding. Thus, when an antibody recognizes an antigen, it binds to it, potentially blocking its action and marking it for attention by other aspects of the immune system.

Proteins and polysaccharides are large complex macromolecules, and the number of different ones in all of nature is vast. This means that the total number of antigens, both self and nonself, that are potentially recognizable by the adaptive immune system is extremely large—far larger than the number of connections in the brain. So, for many years the central question of immunology was, how can antibodies (and T-cell receptors) be produced that specifically recognize any conceivable foreign protein or polysaccharide that might come along when the number of such structures is effectively infinite?

Part of the answer that has unfolded over the past thirty years is that your body produces vast numbers of different antibodies. At any time, your blood contains millions of antibodies, each with a different binding specificity. To understand how this is possible without having millions of different antibody genes encoded in your DNA and how your body can produce antibodies that specifically bind to antigens it has never before encountered, we need to explore the structure of antibodies and the nature of the cells that produce them.

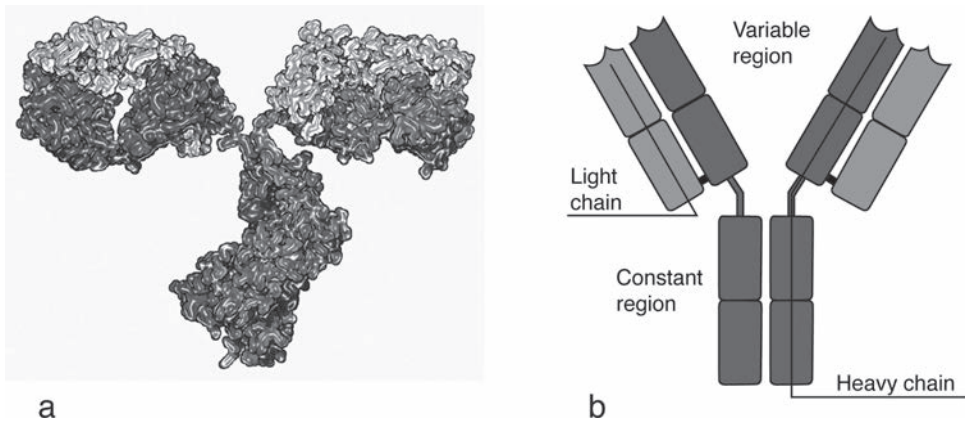
The immune system is a disperse organ made of leukocytes (white blood cells) scattered throughout the body. B-cells, T-cells, and macrophages (amoeba-like cells) are all leukocytes. There are a number of other cell types we will not discuss. Each cell type has distinct roles in

body defense. A key principle to keep in mind when trying to understand the molecular basis of immunity is that *each B-cell produces many copies of just one antibody* (i.e., many copies of the same protein design). Each T-cell also produces just one kind of T-cell receptor. This means there are as many different B-cells and T-cells as there are different antibodies and TCRs. Our focus will be on B-cells.

The structure of all antibodies includes a molecular face with the potential to bind to some specific antigen because the antigen and the antibody surface fit precisely together, much the way a key fits a specific lock. These antigen-binding surfaces differ from antibody to antibody and determine antibody specificity. In order for the system as a whole to recognize foreign antigens never previously encountered, it must produce a large number of different antibodies, each with different antigen-binding sites. Because each B-cell produces only a single antibody design, large numbers of different B-cells must be produced, each characterized by a unique antibody structure. The primary strategy is simple enough. If a large number of different B-cells (and corresponding antibodies) are produced, then by pure chance some of them will recognize (bind to) some antigen on the surface of a foreign invader. Because foreign invaders are living things made of proteins and other biological molecules, and they must interact with the environment outside themselves, they cannot avoid having unique molecular features on their surfaces. Each can potentially be identified by some antibody.

Since the structure of every antibody is determined by its particular sequence of amino acids, which, in turn, is determined by a corresponding sequence of nucleotides in DNA (recall how proteins are made as discussed in chapter 4), different B-cells are characterized by different nucleotide sequences in their DNA (antibody genes). But, how can this be? Every freshman biology student is taught that the DNA in all of our cells is the same. The answer is that this simple dictum is not true for antibody-producing cells. When the DNA of skin or liver cells (any nonimmune system cell) is analyzed, there are no antibody genes, just gene fragments. Only B-cells have complete antibody genes. During the maturation of B-cells, new antibody genes are created by (randomly) combining preexisting gene fragments. This is the first-level explanation of how so many different antibody protein sequences can be generated by the same body. Each newly produced B-cell has a unique antibody gene.





*Figure 7.3.* Panel a: a space-filling model showing the three-dimensional shape of an antibody at the atomic level (Copyright, RSCB Protein Data Bank, credit to David S. Goodsell). Panel b: schematic antibody structure in block diagram form. The antigen binding variable regions are at the ends of each arm.

Actually, antibodies consist of four protein subunits (four amino acid chains), two large subunits called heavy chains and two small subunits called light chains. The two heavy chains are identical, and the two light chains are identical. Because of this, every antibody is determined by two genes, one for heavy chains and one for light chains. So, each B-cell has two unique genes that contribute to the antibodies it produces. Antibody molecular structure is shown in figure 7.3. A B-cell synthesizes many copies of the same heavy chain and the same light chain, and the two chains combine to form many copies of the same antibody.

Physical and chemical comparison of many antibodies shows that all have a constant region and a variable region. The constant region is always the same within an antibody class, while the variable region differs from antibody to antibody. It is the variable region that binds to antigens. Both the heavy and light chains contribute to the variable region. Because all heavy and light chains produced by a particular B-cell are the same, all antibodies produced by that cell will bind to the same antigen.

During the maturation of B-cells in the bone marrow, the DNA in each future B-cell is rearranged so that one heavy chain gene and

one light chain gene are assembled from gene segments. Light chain variable region segments come in two types, V (variable) and J (junction), while heavy chain variable region segments come in three types (V, J, and D). Non-B-cell human DNA contains some seventy functional light chain V segments and nine J segments. The molecular decision determining which V and which J segments are joined together is largely random, and only after a functional rearrangement of the DNA has been completed for both heavy and light chains does a B-cell produce antibodies. These DNA rearrangements potentially generate 320 different light chains and 6,000 different heavy chains. By randomly matching one light chain (320 possibilities) with one heavy chain (6,000 possibilities), 1.9 million different combinations of heavy and light chains can be produced. This means that roughly 2 million different antibody genes can be produced as a direct result of the DNA rearrangement process. This number is impressive but is only a tiny fraction of the antibody diversity actually observed.

Part of the explanation for the greater diversity of antibodies is that DNA rearrangements occurring in B-cells are not very precise. In fact, they are downright sloppy. Special enzymes randomly add and remove nucleotides at the ends of the cut segments before V and J regions are joined. Since the sequence of nucleotides in DNA determines the sequence of amino acids that constitute each antibody, this random alteration of nucleotide sequences at the junction between V and J results in a significantly larger number of distinct antibodies than would be produced by precise DNA joining. A price paid for this intentionally sloppy mechanism is that many (more than two-thirds) of the antibody genes produced are nonfunctional. B-cells that code for nonfunctional antibodies have one of two fates: they either undergo a second round of DNA rearrangement or they die.

Even more variability is introduced if a B-cell should happen to recognize a foreign antigen. Vast numbers of immature B-cells are produced every day from stem cells located in the bone marrow. Formation of antibody genes by DNA rearrangement occurs in these immature B-cells. When rearrangement is completed, newly synthesized antibodies appear on the cell surface. The fate of these cells is determined by what, if anything, the antibodies bind to. If they bind to antigens present within the bone marrow environment (self-antigens), the cell dies or becomes unresponsive. These cells are eliminated from the B-cell

population. They are said to be clonally deleted because they have no descendants. B-cells producing functional antibodies that do not bind to self-antigens are released into the blood.

The life span of a circulating B-cell is rather short, only a few days, unless the antibodies displayed on its surface bind to a foreign antigen. If this should occur, and the presence of a foreign agent is confirmed by simultaneous T-cell activity (discussed later), then the B-cell becomes activated. Activation entails a variety of changes to the cell. It begins to divide, hypermutation is activated, and antibodies that were previously bound to the cell surface are released to circulate freely in the blood and lymph. Ultimately, two types of progeny are produced: blast cells that secrete vast numbers of freely circulating antibodies, and long-lived memory cells that continue to display antibodies on their surfaces. Memory B-cells provide the basis of long-lasting humoral immunity. These cells are rapidly activated upon reinfection by a previously encountered disease organism.

For many years a puzzling aspect of humoral immunity was that antibodies produced later during an infection are more specific and bind more tightly to foreign antigens than those produced early in the infection. The reason for this is now known to be hypermutation and selection. When a B-cell is activated, it replicates its DNA in preparation for cell division. Special enzymes are produced that cause the replication of variable region antibody gene DNA to be error prone. Other cellular DNA is not affected by these enzymes. The rate of nucleotide sequence mistakes occurring in the variable regions of both heavy and light chain genes is estimated to be about one error in every thousand nucleotides ( $10^{-3}$ ), while the normal error rate for DNA replication is closer to one in 100 million ( $10^{-8}$ ). Given the size of the heavy- and light-chain variable regions, one of these errors occurs in heavy- or light-chain DNA replication roughly every second cell division. Each error results in a cell that produces a modified antibody. Because the changes occur in the variable region, the antibodies have different binding specificities. Most bind less well, but some bind better to the target antigen. The fate of activated B-cells is determined by how tightly their antibodies bind.

Antibody binding is not an all or none proposition. Some antibodies bind to more than one antigen, and any one antigen can be recognized by many different antibodies. Some antibody-antigen interactions are strong, while others are weak. B-cells are only activated if their antibodies

bind at least moderately strongly to a foreign antigen. During B-cell clonal expansion (time of rapid cell division), hypermutation results in B-cells synthesizing antibody variants similar to, but different from, the original antibody that recognized the foreign antigen. Most of these variants do not bind as well as the original, and these B-cells are no longer stimulated to divide. Some of the variants, however, bind better than the original, and these are stimulated to divide even faster. In time the better binding B-cells outgrow the weaker binding cells. Hypermutation continues during the entire phase of B-cell expansion. So, over a period of days and weeks, better and better antibodies are discovered by random mutation, and their corresponding B-cells are stimulated more strongly to divide than those producing more weakly binding antibodies. The B-cells producing antibodies that bind most tightly to foreign antigens are stimulated to divide most rapidly. These B-cells come to dominate the population of B-cells circulating in the blood, and the antibodies they produce become the most common antibodies in the blood. This system of restricted random modification followed by selection automatically generates and then improves the antibodies it produces.

Hypermutation and clonal selection constitute nothing less than evolution within the body. It is survival of the fittest B-cells, with fitness being determined by how tightly antibodies bind to foreign antigens present. By this simple method, the body can produce antibodies that bind tightly and specifically to almost any molecular structure because appropriate antibodies are evolved for each new antigen. Every new infection results in the evolution of new antibodies appropriate for identifying and destroying whatever pathogen is a problem at the current time.

All the elements of an evolutionary process presented in chapter 5 (see the section “Where do instructions come from?”), and all the features of the engine of complexity shown in figure 5.2 are present in this body system. Heritability is based on DNA, populations of cells evolve, random variation of inherited information is generated, and cells with certain sequences are selected to survive and reproduce, while others are eliminated. The immune response really does illustrate evolution in the body. New cells are evolved in a week or two that with high specificity target any invading microorganism.

B-cells and the antibodies they produce constitute only one aspect of the immune system. Circulating antibodies coat the surfaces of

foreign organisms interfering with their function and marking them for destruction by macrophages, amoeba-like cells that patrol body. Because antibody binding has such powerful effects, the production of antibodies that bind to the body's own cells and proteins can be quite deleterious. As a check, B-cell activation (usually) requires co-stimulation by T-cells. The double signal required to stimulate B-cells (antibody binding and stimulation by T-cells) helps ensure that large numbers of self-reacting antibodies are not produced.

T-cells come in a variety of subtypes including T-helper cells and T-killer cells; and in a manner similar to B-cells, T-cells produce specialized proteins located on their surfaces called T-cell receptors (TCRs). TCRs also recognize foreign antigens. Like antibodies, T-cell receptors have variable and constant regions, and it is the variable regions that bind antigens. Also, like for antibodies, DNA in non-T-cells does not have intact TCR genes; they too are assembled from TCR gene fragments in a manner similar to antibody genes.

T-cells also originate in the bone marrow by cell division of hematopoietic stem cells. Immature T-cells migrate to the thymus gland where DNA rearrangement occurs, creating functional TCR genes from V, J, and C gene fragments. Like antibodies, the joining process includes the random addition and deletion of nucleotides in the joint regions creating a huge variety of TCR proteins each having a different specificity, and also like B-cells, mature T-cells are selected from a much larger pool of immature T-cells.

The antigens recognized by TCRs differ from those recognized by antibodies. While among the millions of antibodies produced there are some that can bind to virtually any molecular feature of the appropriate size, TCRs bind only to polypeptides (short chains of amino acids) that are simultaneously bound to cell surface proteins called MHC (major histocompatibility) proteins<sup>4</sup>. Nearly all cells in the body break down a small fraction of the proteins they produce and combine the resulting fragments with MHC proteins. These peptide-MHC complexes are transported to the cell surface where they are "displayed" on the cell surface. Thus, most cells in the body are studded with displayed fragments representing all the proteins produced inside them.

T-cells are constantly scanning these presented peptides for foreign ones. When a non-self-antigen is found, what happens next depends on the nature of the cell doing the presenting. For most cells, the presence

of a foreign peptide antigen that is presented on its surface signals that the cell is infected with a virus or bacterium. Generally, the fate of such cells is to be killed, eliminating the invader's home in the body. B-cells behave differently. Immature B-cells have antibodies on their surfaces. When those surface antibodies bind to a foreign antigen, the antigen is internalized, fragmented, and the resulting fragments redisplayed on the cell surface. If these "presented" peptide fragments are recognized as foreign by T-cells, the presenting B-cell receives a second signal (primary antigen binding is the first signal) confirming that it has indeed encountered foreign material. Generally, this double stimulation is required before B-cells are activated to undergo rapid cell division and maturation into blast cells secreting vast numbers of antibodies.

Because T-cells regulate the immune response, it is very important that T-cells do not recognize self-antigens. Developing T-cells also undergo an intense phase of selective elimination during their maturation. Cells that survive selection are released from the thymus and populate the spleen and lymph nodes. TCR gene formation does not involve a hypermutation phase, but it does include a wider range of random changes made during the joining process. The overall result is similar to clonal selection of B-cells.

As with B-cells, the production of T-cells highly specific to foreign antigens involves the generation of large numbers of cells having different TCRs, each produced by a genetically distinct cell followed by intense negative selection eliminating cells with unwanted binding capability and positive selection by growth stimulation amplifying those (initially few) cells that specifically identify proteins that should not be in the body. B-cell maturation goes one step further by employing repeated rounds of mutation and selection to evolve ever-improving specific responses to each invading microorganism on a time scale faster than most microorganisms can themselves evolve to circumvent the defense. The B-cell response fully engages the engine of complexity, and this explains how it can be so finely tuned to each new disease.

## What does copying add to selection?

Development of the human body from a single cell provides many examples of the structural richness that is possible when the repeated

production of random variation is combined with nonrandom selection. All phases of body development from embryo to adult exhibit random activities at the cellular level, and body formation depends utterly on the new possibilities generated by these activities coupled with selection of those outcomes that satisfy previously built-in criteria. Always new structure is based on old structure, and at every stage selection favors some cells and eliminates others. The survivors serve as the progenitors for new cells that undergo further rounds of selection. Except in the immune system, cells and extensions of cells are not genetically selected during development, but rather, are positionally selected. Those in the right place that make the right connections are stimulated, and those that don't are eliminated. This process is much like sculpting. A natural consequence of the strategy is great variability from individual to individual at the cell and molecular levels, even though large-scale structures are quite similar.

The creation of new B-cells differs from this strategy in one very important respect. During brain development the DNA sequences that code for proteins that create neuronal processes do not change. When axons and dendrites grow and explore specific neighborhoods, the pattern of gene expression changes, but the underlying instructions do not. The DNA directing the growth of a favored process is the same as the DNA in one that withers away. In contrast, during an immune response, the DNA in specific cell lineages does change. For B-cells, DNA sequences that code for antibody variable regions change permanently and uniquely in each cell lineage. By allowing the instructions to change, the system has the maximum potential of adapting to any conceivable intruder. Brain development, though impressive, exhibits more limited creativity. If during brain development neurons were selected genetically as well as positionally, human brains wouldn't just differ in microscopic detail, but in all kinds of other ways. In such a world, brains wouldn't be species specific! This wider-ranging kind of change does happen during the evolution of species (over millions of years) but not during the formation of individuals.

Changes that occur when the immune system responds to foreign antigens mimic the evolution of organisms in nature, but within the body in a few short weeks. The body cannot possibly "know" in advance what kind of cellular or viral assault it will next encounter.



Copying with random minor change (mutation) followed by selection-favoring variants that do a better job provides an endless source of new ways (new protein structures) for solving the problem at hand. Any preprogrammed menu of possibilities could be exploited by fast-evolving microorganisms. Hypermutation combined with clonal selection assures that whenever weakly binding B-cells are stimulated, highly specific antibodies will soon be produced.

One reason computations based on linearly encoded information are so versatile and powerful is ease of copying. It is easy to copy one- and two-dimensional objects, but very hard to directly copy three-dimensional ones. This is a consequence of geometry. It is why you can buy a copy machine but not a Star Trek replicator. When you have something useful, it is often the case that more of them are more useful. It is also the case that when slightly modified, some of the new somethings are even more useful. Most things in the physical world are three-dimensional and so can't be (directly) copied, but because one-dimensional instructions can be used to produce multiple objects having the same three-dimensional design, instructions provide a way around the three-dimensional copying impasse. This simple principle is the basis of manufacturing. Less obvious is the aspect of nature that minor modifications to instructions frequently result in new, but also useful, things. This is what ultimately makes commercial product development, plant and animal evolution, and adaptive immunity all possible.

By exploring random, but minor, changes in antibody structural detail on a massively parallel scale, the body is able to generate appropriate antibodies to combat any invader made of proteins. Since all earthy life is based on proteins, this is a very reliable strategy. In brain development the formation of random connections provides an efficient way to explore a complicated local environment. But, unlike the situation of antibody production, the DNA-based information needed to produce axons and dendrites doesn't change in one's lifetime. Subsequent negative selection of those neuronal connections that don't meet predefined criteria makes it possible to construct a very complex and more-or-less reproducible wiring pattern from an initially chaotic situation without requiring an impossible amount of previously stored information. One consequence of this strategy is that the brain isn't functional "off the shelf," like a store-bought computer, a learning process must follow for it to work. We will explore this further in chapter 10.



# 8

---

## Taking Control of the Cycle

Can the engine of complexity accept any form of information?

The formation of body and brain during human development is impressive. It demonstrates the creative potential of a physical system capable of generating variations, selecting some, and building new structure on that selected. But this strategy does have limits. There are many things in the physical domain that cannot be achieved so simply. This is why instruction use is so important in our world. Recall that essentially anything in the physical domain is achievable if the proper instructions can be obtained and employed. The engine of complexity operates at the level of instructions. By applying to instructions the basic principle of building on selected variants, it becomes possible to generate new instructions, and the door is opened to endless possibilities. Three examples we discussed in earlier chapters are: life, disease-specific immunity, and computer programs that evolve workable answers to otherwise intractable problems. In later chapters I will argue that human technology and human thought are

also outcomes of the same computational strategy. But, the issue we will turn to now is the extent to which the engine of complexity can be taken over, controlled, and manipulated by human engineers for human benefit.

Every engine of complexity computation has three basic elements: (1) a body of information encoded in such a way that it can be copied; (2) a copying mechanism that introduces limited changes (often nothing more than errors in copying); and (3) a means of choosing which of the copies are copied again (diagramed in figure 5.2). We need to ask if there are informational structures not amenable to modification by the engine of complexity. It turns out that this is no different than asking if there are structures that cannot be copied.

In biology every organism has encoded within its cells information that is used to specify many aspects of its structure. This information resides in sequences of nucleotides that constitute DNA molecules. DNA and its encoded information are copied whenever a cell reproduces. The system is capable of encoding quite a lot of information because a very large number of different nucleotide sequences are possible and DNA molecules are very large (as molecules go). This same encoding system, but a different way of evolving it, provides the basis of our adaptive immune system. Computer algorithms employ a dizzying array of structures to encode information, but all of them are ultimately reducible to sequences of 0s and 1s. Human thought and human culture are also based on encoded information. In these cases the information is encoded in neuronal connection patterns occurring in the brain.

Earlier, we saw that whenever a system can (potentially) take on multiple configurations, there is information associated with each configuration. What makes information so useful in biology and human culture is that it can *represent* something else. This allows for meaning. By assigning specific meanings to specific configurations it is possible to encode instructions for making something else. The variety of informational structures observed to encode information raises the question of whether there are any encoding schemes *not amenable* to engine of complexity-based change. The central role for copying does seem to impose limits. Geometry matters. Generally speaking, one- and two-dimensional structures are amenable to copying while three-dimensional structures present problems.

Copying can be accomplished in at least three ways. It can be direct, such as when a T is copied as T and an A as A, it can involve an intermediate template, or it can involve instructions. Direct copying is not common in nature or in technology. An office copy machine utilizes a template; each printed page is produced from a mirror image created on a special roller. Later we will see that when DNA is copied, each strand acts as a template for a new strand. In this scheme, every A nucleotide on the old strand directs the incorporation of a T nucleotide on a new strand, and likewise every T an A, every G a C, and every C a G. By this simple means the sequence of each new strand is always determined by the sequence of a template strand. Many industrial processes involve the use of a mold to form a desired shape. Molds are templates that allow the copying of simple three-dimensional shapes such as plastic bottles or bronze statues. Molds do not work for complex three-dimensional objects. Copying of complex objects cannot be done directly or even with the use of a template. If copies of such objects are to be made, a one- or two-dimensional representation of the three-dimensional object must first be made and the representation used to direct a process that creates multiple three-dimensional objects. The process can be amplified by making additional copies of the representation. Instructions are one-dimensional representations of methods that can be used to create copies of three-dimensional objects; blueprints are two-dimensional representations having this same purpose.

An example we discussed earlier is a protein with a specific structure. Proteins play a central role in the function of cells and of the body. Without proteins there would be no life as we know it. The function of every protein is a direct consequence of its shape. To make many copies of a protein, the cell makes a one-dimensional representation in the form of messenger RNA, then translates this representation into a chain of amino acids that folds spontaneously into a new copy of the desired form. Each mRNA sequence always results in the same protein shape. When DNA is replicated, or messenger RNA synthesized, protein representations, that is, gene sequences, are copied, but a folded protein is never copied directly in three-dimensional form.

Human technology works in much the same way. An automobile is not duplicated from another automobile; rather, identical cars are made according to the same plan. As long as the plan is the same, identical automobiles are produced. If a modification is desired, the plan

must be modified before a new model is produced. This is the magic of instructions; they allow the reproduction of complex (three-dimensional) structures that are not amenable to direct or template copying. Representation of structure in the form of one-dimensional instructions makes it possible to make copies of any object that can be built.

Because information present in three-dimensional shapes can be represented by information encoded in one dimension, in principle there are always one-dimensional representations (instructions) that can be copied and modified and then transformed back into new three-dimensional objects. This transformation, modification, and retransformation cycle underlies both life and human technology. So, the answer to the section question is a qualified no, not all forms of information can be directly or even template copied, and those that cannot also cannot serve as input/output to the engine of complexity. However, it does seem that all structures can be represented by one-dimensional structures that can be copied and manipulated and then used to create new “copies” of three-dimensional structure. Thus, a human body cannot be copied, but human DNA can be, and human DNA when placed in the proper environment (a human egg cell) can produce a new body (through the process of human development). By copying the DNA and activating the proper sequential expression of genes in the proper cellular environment, it is possible, in principle, to create multiple copies of the same human form. Hollywood calls them clones. Such experiments are currently outlawed for human DNA in most countries of the world, but are commonly performed with DNA of other organisms.

### Can variation be controlled?

The engine of complexity is a strategy for manipulating information that is based on creating variation and then culling nonrandomly. Variability in outputs and outcomes provides options to choose from. The simplest way to introduce change during a copying process is to allow errors to occur. At the molecular level, errors are typically random or nearly so. This is because when change is due to the thermal motion of molecules there is no way such a change could in any way be dictated by some future need of the system. This observation, however, does

not mean the engine *requires* random changes. The engine of complexity computation works just fine when changes made during a copying process are not random. Any source of variation can provide grist for selection to occur. Remarkably, though, random change affords the maximal long-term opportunity. Whenever change is not random the system evolves, but future potential is restricted. When there is a preconceived notion about what the future should be, all sorts of unimagined possibilities are ruled out. Only when change includes a random element are the opportunities available to the system truly open ended. Random change maximizes future potential.

It is hard to think about information changes in the abstract, so let us examine in some detail various ways that one very important evolutionary system, life, introduces variation. The ultimate sources of new heritable variation in biology are mistakes and catastrophes, but life employs some other tricks that allow a full exploitation of this source. The molecular machinery that duplicates DNA is quite sophisticated but not error free. Also, DNA molecules are very long and fragile, and many chemicals and most forms of high-energy radiation can damage DNA. Because of this the information encoded in your DNA is under constant assault. Cells protect themselves by producing enzymes that are good at detecting and repairing damage, but the repair mechanisms are not perfect. This is actually a good thing; errors are essential if continued evolution is to occur. If a species were to develop mechanisms that achieved perfect DNA replication and repair, it would evolve no further. Eventually, its environment would change; other evolving organisms would change in response to the new conditions and out-compete it for food and other resources. If this were to happen, the previously perfect organism would be at increasing disadvantage and would go extinct. With no errors, there would be no change; and in biology, unchanging perfection is a ticket to extinction.

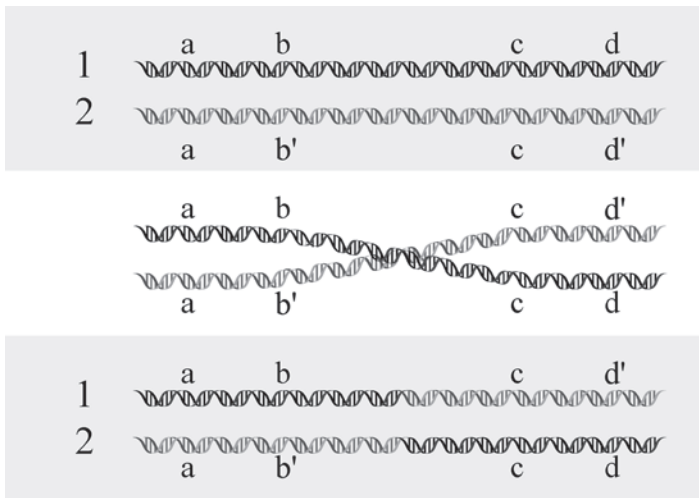
One remarkable consequence of the critical role of errors is that over millions of generations, organisms have evolved their mutation rates. The engine of complexity works best over a narrow range of mutation rates that provides a good rate of change without excessively disrupting the already accumulated body of useful information. Too much change is as bad as too little. This was illustrated in figure 6.4, for the one-max computer algorithm where the optimal mutation rate for creating a 100-bit string of 1s was 2 percent per bit per generation. It is also

illustrated by mutation rates observed in nature. A virus with a genome of 10,000 ( $10^4$ ) nucleotides will tend to have a mutation rate of about  $10^{-4}$  mutations per nucleotide per generation, while a bacterium with 4 million nucleotides ( $4 \times 10^6$ ) will have a mutation rate closer to  $10^{-6}$ , and a mammal with 3 billion nucleotides ( $3 \times 10^9$ ) in its DNA, a mutation rate of about  $10^{-8}$ .

Another type of change to DNA that turns out to be important in the evolution of life is duplication. When a gene or a block of genes is duplicated within the same DNA molecule, no information is lost. Duplication allows future changes in one copy without disrupting the original function. Analysis of plant and animal genomes reveals that duplication followed by evolutionary change in the duplicated DNA has repeatedly led to the expansion and diversification of protein functions available to cells. Deletion (loss) of sequences is another, but more disruptive, source of change.

In biology two additional and rather different mechanisms assure that variations initially created by mutations, duplications, and deletions “fit in” with all the other gene activities that make up an organism and assure that useful new sequences are fully exploited. These are random assortment of chromosomes and recombination of DNA during meiosis, the process that produces sperm and egg cells. These mechanisms are important because it is not just the outside environment that determines whether or not a DNA variant propagates or disappears but also internal interactions with other genes (and gene variants) in the same organism. These mechanisms in the context of sexual reproduction ensure that every new offspring obtains a complete set and also a unique mixture of gene variants from their parents. Recombination allows mixing and matching of information while random assortment guarantees that each offspring has gene variants from all four grandparents.<sup>1</sup>

Random assortment of chromosomes mixes sequence variations already present in a population. All multicellular organisms have multiple DNA molecules in their cells; and each DNA molecule is bound up with proteins and RNA to form a microscopic structure called a chromosome. Short DNA molecules make short chromosomes, and long molecules make long chromosomes. Because DNA molecules in a cell have different lengths it is possible for a laboratory worker to identify the chromosomes present by their size. In addition, most



*Figure 8.1.* Each horizontal line represents a DNA molecule (one from each parent) and the letters above each line a gene encoded by a sequence of nucleotides that are not shown. Because we, and most other creatures, have two parents, we have two copies of each gene and these two copies are often slightly different, indicated by a prime ('). During recombination, two DNA strands carrying the same genes are lined up, each is cut at the same location, and the ends rejoined with the paired strand.

multicellular organisms have two nonidentical copies of each DNA molecule in every cell. This means that every cell has two of each kind of chromosome. One of each pair came from the father and the other from the mother. Human cells have forty-six chromosomes, two each of twenty-three different kinds. Guinea pig cells have sixty-four chromosomes, fruit flies have eight, while corn (maize) has twenty. The DNA in each chromosome encodes a particular set of genes. During sexual reproduction, gametes are produced that have only one of each kind of chromosome, but *which one* (from mother or father) is determined randomly. Because of this, on average, one-fourth of each organism's chromosomes derive from each grandparent. But, this is only an average; the actual distribution is random.

Recombination is illustrated in figure 8.1. In its most common manifestation, known as homologous recombination, two DNA molecules (1 and 2 in the figure) containing similar but not identical sequences

**Table 8.1**

Major sources of heritable change in natural biological systems.

Category of change	Nature of change
Replication errors	Different nucleotides in existing sequences
Repair errors	Different nucleotides in existing sequences
Segmental duplication	Duplication of an existing nucleotide sequence
Segmental deletion	Lost of nucleotide sequences
Segmental rearrangement	Changing the order of DNA sequences within a DNA molecule
Recombination	New combinations of existing DNA variants
Random assortment of chromosomes	New mixtures of grandparental chromosomes and corresponding DNA sequences
Horizontal gene transfer	Complete new genes from other organisms (can be natural or human engineered)

*Note:* Each change listed affects DNA, and each involves random change.

are paired up so that like sequences are juxtaposed; both molecules are then cut at the same randomly chosen point and reconnected in such a way that each left side of the break point is derived from one parental molecule and each right side from the other.

The combination of random assortment and recombination guarantees that each offspring derived from the union of egg and sperm has a unique combination of gene variants and two of each gene kind. This, in turn, guarantees that every gene variant gets “tried out” in combination with variants of other genes over time. Those DNA sequences that provide advantage in many genetic contexts will tend, over time, to spread widely through the population; those that do not will disappear. All the time this is happening, new variants are constantly being generated by errors in replication and repair, and more rarely, by duplications and deletions. If the introduction of new variation is too rapid, the system can become overwhelmed by errors and collapse from excessive loss of (useful) information. If the introduction rate is too small, change will be very slow.

Table 8.1 lists the major sources of variation in biological systems. Recent advances in molecular biology have enabled new means of



introducing variation. This capability is collectively called genetic engineering. It is now possible to move genes (DNA sequences) from one organism to another, and it is possible to synthesize new genes in the laboratory and introduce them into an organism's chromosomes. Such activities can change specific aspects of the total body of DNA-based information present in cells and cause the resulting organisms to have new properties. The introduction of preplanned human-designed variants is completely compatible with the engine of complexity cycle. From the computational perspective, human tinkering simply provides another source of variation.

Evolutionary computer algorithms provide a laboratory for demonstrating the spectrum of change mechanisms that may be incorporated into evolutionary systems. Algorithms have been written that imitate all the natural methods listed in Table 8.1 and others that are unique to the computer environment. Generally, it has been found that a combination of random change at the single-bit level and recombination provides adequate variation for solving most computer problems that are encountered.<sup>2</sup> Computer experiments and recent advances in genetic engineering demonstrate convincingly that if technology can provide a way, there are no theoretical barriers to the artificial creation of new variation in evolutionary systems.

### Must selection be natural?

Much has been written about the central role of natural selection in biological evolution. It is interesting that Darwin himself was persuaded that his grand idea was essentially correct by observing the accomplishments of plant and animal breeders: in other words, by observing the power of nonnatural selection. The variety of dogs, horses, and pigeons that had been produced by animal breeders played a key role in the arguments he presented in his classic book "*On the Origin of Species by Means of Natural Selection.*"<sup>3</sup> It is quite clear that the engine of complexity cycle works very well when selection is not "natural." Scientists studying bacterial genetics routinely produce bacterial strains that are able to grow in the presence of a toxic chemical or live and grow using nutrients that parental strains are unable to utilize. The standard procedure for creating new strains of bacteria involves taking control of

selection. A small laboratory culture contains more than a billion free living bacterial cells. Because of mutation, there are many differences even if the culture originated from a single cell. With a mutation rate of one change per generation per individual, a billion bacterial cells carry more than a billion differences in their collective DNA sequences, this despite the fact that many mutations were deleterious and vanished from the population.

The trick to selecting a strain of bacteria exhibiting a new trait is growing the culture under conditions that allow limited growth for normal or “wild type” cells and that allow bacteria with the desired trait to grow faster—just a tiny bit faster will do. For example, if one wants to create a strain of bacteria resistant to penicillin, it does no good to add such a small amount of penicillin that no cells are inhibited or to add so much that all bacteria are killed. Rather, one must add just enough penicillin to kill most of the bacteria but not all of them. Typically, the survivors grow, but very slowly. The reason they are able to grow at all is because they differ in some way from the majority of cells. But, because they grow and divide, they also mutate. Sooner or later one or more of these mutations will enable still faster growth. The faster-growing cells quickly outgrow the more slowly growing cells and come to dominate the culture. Our microbiologist then increases the concentration of penicillin, causing most of these cells to die. But once again the concentration chosen must allow a few to grow slowly. Some (random) mutations occurring in these now slow-growing cells again enable faster growth at the higher concentration of penicillin. By repeating this procedure, a microbiologist can rather quickly create a strain of bacteria resistant to concentrations of penicillin that would have killed all the cells in the original culture a hundred times over. Taking control of selection allows creation of new strains with new properties.

This same process when it occurs unintentionally presents a challenge to modern medicine. Antibiotics are one of the most beneficial achievements of twentieth-century medical science. These chemicals are widely used but are not always taken in the proper doses. Patients too often do not follow the prescription. A consequence may be that not all of the offending bacteria are killed. Those that remain are the most resistant. If these are passed to someone else who also doesn't take the prescribed dose, ever more resistant strains are selected. When

sublethal exposure to antibiotics is repeated over and over, strains very resistant to the antibiotic become widespread. Evolution occurs, and doctors lose an important tool for fighting infection.

This phenomenon is not limited to bacteria: HIV (AIDS virus) evolves resistance to antiviral drugs, malaria parasites to antimalarial drugs, and mosquitoes to DDT. The repeated cycle of mutation followed by selection is an effective mechanism for modifying existing genes but not very effective for creating new genes with totally new capabilities. This is best accomplished by getting the new gene from somewhere else. Living in the presence of antibiotics is best accomplished by expressing genes that destroy the chemical agent. Most antibiotics are isolated from soil fungi, and soil bacteria have been growing in the presence of soil fungi for a very long time. A consequence of this long history is that many soil bacteria have evolved genes that code for proteins that destroy specific antibiotics. Bacteria may be fully susceptible to some antibiotics and very resistant to others that are targeted by their particular set of resistance genes. Most bacteria also have mechanisms for exchanging genes with other bacteria. This exchange allows genes that are useful to be acquired intact. Under the right conditions, gene transfer mechanisms allow genes to spread rapidly through bacterial populations. Quite often, antibiotic resistance genes are packaged together on separate pieces of DNA called plasmids, and these plasmids are passed around from bacterium to bacterium. Because of this, places that use lots of antibiotics, such as hospitals, tend to be populated by bacteria that are resistant to antibiotics. This is a natural consequence of evolution in response to selection and is why it is much more dangerous to get an infection in a hospital than at home.

Modern agriculture is based on human selection layered onto natural selection. Early in the agricultural revolution about 10,000 years ago, human beings learned to co-opt the evolutionary process. During early forays into agriculture, it was probably natural for farmers to eat the best seeds and save the worst for next year. A few, however, must have realized an advantage to selecting next year's seeds from the *best* of this year's crop and by breeding the *best* animals in the herd or flock while slaughtering those with less favorable traits. Whoever did not follow this practice would have fairly quickly seen the productivity of their crops decline compared to those who did, and breeding from

the best became standard practice in agriculture wherever it was practiced. Application of artificial selection year after year, generation after generation, gradually led to crops and domestic animals that were far superior for human purposes than their wild ancestors.

Cereal crop varieties not only produce more grain per acre than their wild counterparts but also exhibit a host of other modifications that make them superior for human consumption. These include uniform germination (so they come up when planted), attachment of grains to stalk that is both strong and brittle (so that grain remains on the stalk until harvested, but then can be separated by mechanical action), protective seed coverings that are less tough (to facilitate separation of the seed from the husk and easy milling), and plant morphology that is more upright (to facilitate harvesting). Parallel changes have occurred in domestic animals. Domestic animals are generally larger, meatier, more docile, and breed more readily than their wild relatives.

In the past century, knowledge of genetics and the application of modern statistical methods have allowed traditional selection practices to be improved upon. As a result, corn (maize) yields have improved fivefold in the United States in only 100 years. An acre that yielded thirty bushels (760 kg) in 1905 now yields 150 bushels (3800 kg). Some of this is due to improved practice, but more than half of the improvement is due to improved genetics. The story is similar for other major crops. The “green revolution” is frequently cited as the main reason famine has decreased in the world even while the human population has mushroomed. Dairy cows producing more than 20,000 pounds of milk per year are now common, but were unheard of 100 years ago. These cows are the result of sophisticated breeding schemes based on vast data sets containing information about milk production, nutrition, and susceptibility to illness for millions of animals. This information allows genes responsible for desirable traits to be traced over many generations and identification of the best combinations of animals to breed. The underlying principle is the same as for traditional agriculture: breed animals with superior genetics with others having superior genetics, but modern techniques are more efficient.

Modern plant and animal breeding employs the very same engine of complexity outlined in chapter 5 with the difference that selection is not left to nature. Plant and animal breeders have taken control of the selection part of the cycle. Experience indicates that almost any

attribute of plants and animals can be modified by exercising consistent control over selection. Exceptions occur when features exhibit no selectable variation and when a desired trait is completely absent, so there is nothing to modify.

Modern biotechnology allows the introduction of entirely new genes into the DNA of organisms. By this means new traits can be made available for selection. The natural evolutionary cycle relies on *random* changes to generate new variation; genetic engineering permits the introduction of *predetermined* variation. The result is greatly accelerated rates of change. Instead of needing hundreds or even thousands of generations to create strains with otherwise hard-to-achieve properties, desirable traits can be created in a few generations. Commercial examples include corn that is not digestible by moth larvae, soybeans that are resistant to glyphosate (an herbicide that kills most plants), and rice containing high levels of vitamin A. In a sense, this technology is simply imitating what bacteria have done for eons of time when they exchange genes that confer antibiotic resistance.

Genetic algorithms running in computers take the control of selection to the ultimate extreme. In all examples of artificial selection of living organisms, human selection simply supplements natural selection. When a milk cow dies of a disease, natural selection is operating; when a bull is selected by the farmer for breeding, artificial selection is operating. When a computer scientist writes an evolutionary algorithm, the rules of selection are completely specified; there is nothing natural about what happens when the program is run, but nonetheless, evolution occurs.

### Is there any aspect of the cycle that cannot be controlled?

The engine of complexity cycle has several aspects (see figure 5.2). It has a body of information encoded in such a way that it can be copied and evaluated according to consistent criteria. In most systems that information is employed as instructions for the creation of something else. Always there must be a copy mechanism, a way for variation to be introduced into some of the copies, multiple copies must be generated, and selection must occur so that not all copies are subsequently copied. The selection procedure cannot be random and must be consistent.

This is not a very complicated set of requirements; and as might be expected, the scheme is found to operate in a variety of venues.

Because the engine of complexity is fundamentally a computational activity, the logic can be implemented in any system that encodes and copies information. This also means that when the nature of the encoding, copying, and selection are known, there is always the possibility of technological intervention that introduces human-designed variations, new copy techniques, or that substitutes new rules of selection. When the information encoded is in the form of instructions, it should always be possible to use the cycle to create new desired outcomes. In short, control is possible.

In the case of human intervention in the biological realm, the end game of biotechnology is complete control of all aspects of the cycle. This starts with the genome (the DNA). Instead of controlling just selection or selection plus change mechanism, ultimate control will be realized when entire genomes can be specified at will. One can imagine a scenario where a genetic engineer types an entire DNA sequence into a computer that directs the synthesis of the total DNA of a cell. These synthetic molecules will then be introduced into a cell that has had its natural DNA removed, and resultant organisms are grown in a laboratory or field. All the key technologies including the ability to synthesize DNA sequences and the ability to introduce those molecules into cells have already been mastered. Already, virus and small bacterial genomes have been artificially created in the laboratory and introduced into cells.

One thing we can be sure of is that whenever organisms are created in this way, if they reproduce, it will be very difficult to protect them from natural selection. We can also be assured that until a time when our knowledge of physiology and ecology is vastly superior to that of today, returning any such artificial creation to the wild will be a recipe for its quick demise. When human-made selection rules are removed and nature's rules reestablished, wild cousins quickly prove their superiority under the old rules.

In later chapters we will explore the idea that the engine of complexity underlies both human learning and social change. If this is true, the major impediment to social engineering and to technologies that enhance human thinking is the lack of a clear understanding of how information is encoded in human brains. If this were understood better,

then it is very likely human ingenuity would be able to devise ways of introducing predesigned changes in the relevant bodies of information or perhaps of devising ways for altering the rules of selection leading to improved brain function and/or improved social institutions. But before we explore these issues, we need to examine another concept of modern science that is becoming more and more useful, the notion of complex systems.

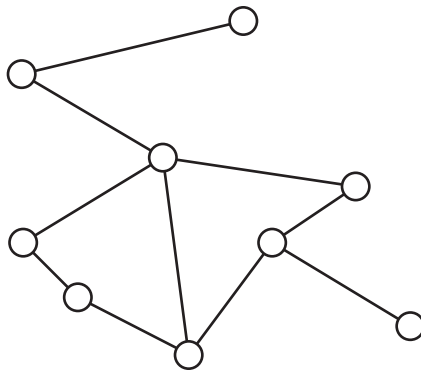
---

## Complex Systems

### What constitutes a complex system?

There is no widely accepted mathematical definition of a complex system. Computers, thunderstorms, the human brain, and corporations are all examples. Always they are made up of many parts interacting with one another in specific ways. Outcomes of these interactions are often hard to predict. Since one of my goals is to explain the existence of complex objects (and there is no clear distinction between an object and a system), it seems appropriate to examine a few such systems in greater detail. Clearly, the opportunity for a system to exhibit complex behavior must be greater when there are more parts. One surprising finding of research in this area is that complicated behavior does not require complicated parts. When systems have lots of interacting parts, even simple parts and simple interaction rules are sometimes sufficient to produce complex structure and behavior. We saw some examples of this with NKS computational systems in chapter 2 and sandpiles in chapter 3.





*Figure 9.1.* A two-dimensional graph with nine nodes and ten edges.

An important aid for visualizing complex systems is the concept of a network. Whenever systems of objects can be characterized in terms of interactions between parts, the systems can be described or, if simple enough, drawn as a network. Mathematicians formalize networks as graphs where individual elements are called nodes and interactions between nodes are called edges. Figure 9.1 shows a simple two-dimensional graph with nine nodes and ten edges.

The mathematical concept of a graph is very general and may be used to describe just about anything that has interacting parts. Graphs can have any number of dimensions (greater than zero). A property of networks is that when the interactions (graph edges) are nonlinear and operate over time, complex behaviors including multiple stable states are common.

Stuart Kauffman built a career by exploring the behaviors of certain kinds of networks, particularly those called Boolean networks because their nodes interact according to the rules of Boolean logic, and by promoting and explaining to others the self-organizing properties of large collections of simple units interacting according to simple rules.<sup>1</sup> One of the principles promoted by Kauffman is summarized by the catchy phrase “edge of chaos.” Large networks tend to be characterized by large domains of parameter space that exhibit chaotic behavior and other large domains where frozen unchanging behaviors dominate; but at the interface, between frozen and chaotic domains, behavior can become complicated—one could even say, creative. Kauffman argues

that evolutionary systems, particularly life, take advantage of such dynamic regions of parameter space. When systems naturally progress toward dynamic states, he calls it “order for free.”

Perhaps because most of his work has been on simple systems, Kauffman tends to downplay the rich behaviors available to systems with complicated parts. Nature is full of situations where systems of simple parts and simple interactions self-create order and exhibit a surprising range of behaviors; but in the realms of life and human activity, we see what is possible when systems have lots of complicated parts. When network nodes are complicated enough to require instructions for their formation, whole new universes of behavior become possible. To illustrate these ideas, we will explore six networks, the last of which provides a definition of life. In each example, characteristic network properties are made possible by careful prespecification of network parts (nodes or edges). The rules of interaction are the laws of chemistry and physics.

## How is an electronic device also a network?

The first examples of networks we will consider are electronic circuits. Electronic circuits are easily visualized; the components (nodes) are physical devices such as resistors, capacitors, and switches; and interactions between the components (edges) occur by the flow of electrons through wires.

### A SWITCHABLE LIGHT

A very simple electronic circuit is one in which a battery is connected in series with a switch and a light bulb (figure 9.2). Closing the switch causes electrons to flow through the circuit. This causes the filament in the bulb to heat up and emit light. Opening the switch halts electron flow, and the light goes out. The network shown has three nodes (battery, switch, and bulb), three edges (wires connecting the nodes), and two stable states (on and off).

Figure 9.3 shows three different circuits with the same components as in figure 9.2. The circuit on the left does nothing when the switch is closed. The center circuit drains the battery when the switch is thrown without doing anything else, and the right circuit drains the battery

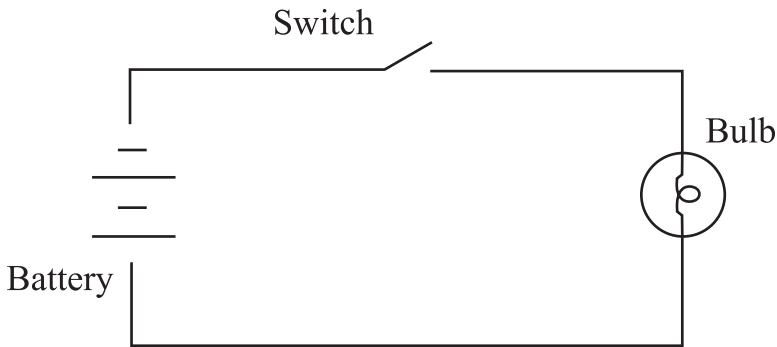


Figure 9.2. A simple electronic circuit with three components and connecting wires shown as straight lines.

whether or not the switch is thrown. None of the three circuits has a state in which the bulb is lit. In general, circuits with one battery, one switch, and one light bulb have the property that closing the switch causes the bulb to emit light only when the three components are connected in series. The number of graphs with these same three nodes that do not have this property is surprisingly large.

### A TRANSISTOR RADIO

Now consider the electronic circuit with fourteen nodes shown in figure 9.4. It diagrams a radio that converts AM radio signals into sound. The network includes the following parts: antenna, coil, diode, transistor, variable capacitor, three fixed capacitors, battery, earphone, three resistors, and wires to connect the parts in a

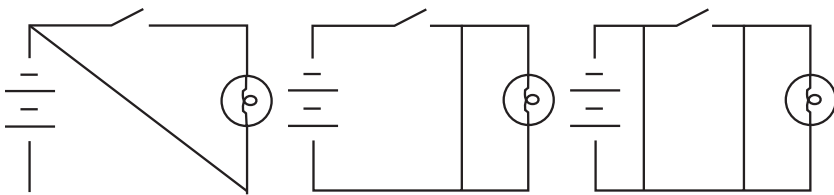


Figure 9.3. Three circuits with the same nodes as the circuit shown in figure 9.2 that do not exhibit notable behaviors.

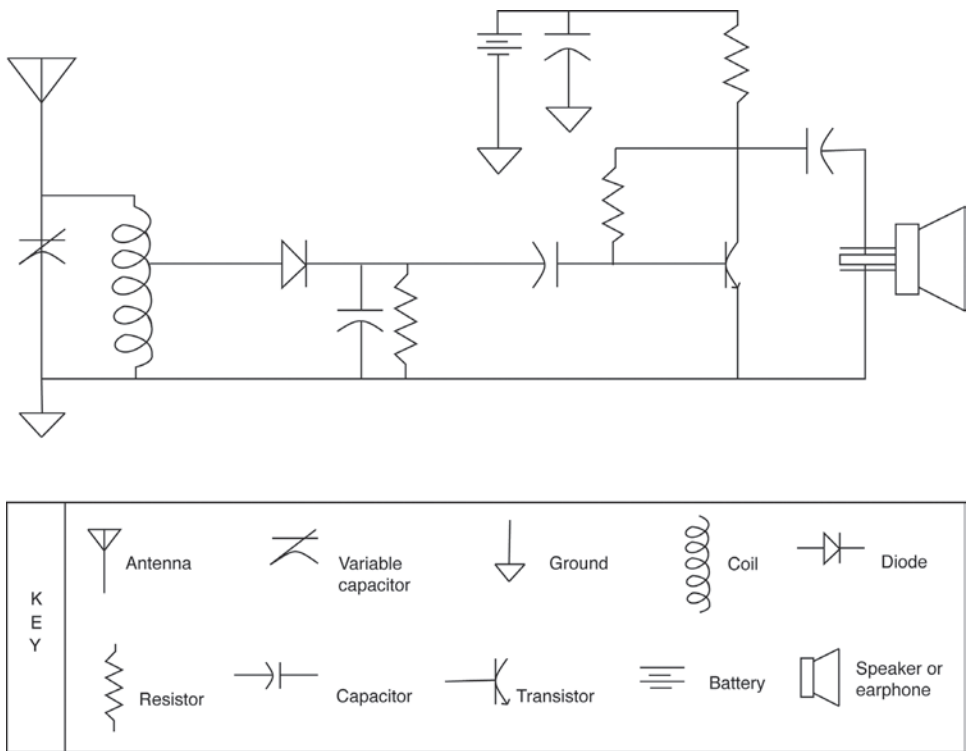


Figure 9.4. Circuit diagram for a simple transistor radio. Ref: <http://www.techlib.com/electronics/crystal.html#Crystal%20Radio>.

particular configuration. Electrons flow from the battery through the transistor or to the earphone.

If the current through the earphone varies rapidly over time, sound will be produced. The variable capacitor and the coil allow an operator to tune the circuit to the frequency of a broadcast signal picked up by the antenna. The transistor circuit amplifies weak electrical oscillations in the antenna so they are large enough to produce audible sound in the earphone.

The output of this network is stable but varies according to whatever signal is captured from the antenna. The network has a large number of stable states determined by the variable capacitor settings. The same fourteen parts can be connected together in a very large number of different ways, but most do not function as a radio. There are other

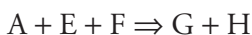
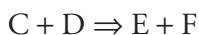
circuits using other parts that also function as radios, but the number of non-radio electronic circuits is uncountably large, and the number of AM radio circuits is vastly smaller. Because of the unfavorable ratio of radios to non-radios among all possible electronic circuits, we do not expect to find radios in nature unless there exists some mechanism for specifying particular parts and their connections.

In contrast to radios, electronic circuits that produce intermittent light are encountered in nature. The circuit diagram of a lightning bolt is very similar to figure 9.2 with charge separations in clouds substituting for the battery and pathways through air substituting for switch, wires, and bulb. As air is heated by a flow of electrons it becomes more electrically conducting (the switch), and when air is heated enough, it becomes incandescent (the light). The difference between electronic circuits found in nature and most of those created by human beings is complexity. As far as we know, human beings are responsible for all complicated electronic circuits that have behaviors significantly more complex than simple oscillations or the turning on and off of lights. All human-built electronic devices require instructions for their creation.

We now turn to several examples of purposeful networks that occur without human intervention, but which nonetheless exhibit complex behaviors.

### What is a biochemical network?

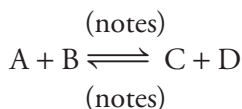
Chemical reactions are traditionally represented by equations of the following form:



Each letter stands for a distinct chemical species, or type of molecule, such as water ( $\text{H}_2\text{O}$ ), ammonia ( $\text{NH}_3$ ), benzene ( $\text{C}_6\text{H}_6$ ), or whatever. A chemical equation tells the reader that one A molecule and two B molecules might combine to form one C molecule, but it does not tell the reader whether or not the reaction will actually occur or how fast it will occur. To predict what will actually happen, a chemist needs more information. An important additional piece of information is the

concentration of each reaction component. A fundamental principle of chemistry is that if a reaction such as  $A + B \Rightarrow C$  can occur, then so can the reverse reaction,  $C \Rightarrow B + A$ . The actual direction is determined by the relative concentrations of the three reaction components and underlying energy relationships. When given such information, a chemist can know the direction of a chemical reaction but, even then, not how fast it will proceed.

Determining the rate of a reaction is more complicated. It depends on the temperature, a property known as the energy of activation, the concentrations of all the reaction components, and the geometry of each chemical species involved in the reaction. Even molecules that participate only indirectly in a reaction can influence the rate. A common way to deal with all these complicating factors is to write a chemical reaction in the following way:



Where the word *notes* above the right-pointing arrow provides the reader additional information that is important for understanding (and predicting) the forward reaction and the word *notes* below the left-pointing arrow provides information important to the reverse reaction. The most commonly used note is a “rate constant,” which combines several parameters into a single number; but the notes may also provide other information such as whether a catalyst (reaction accelerant) is present, special solvents, or anything else that might be important for understanding reaction behavior.

A chemical reaction formula is a one-dimensional graph. When several reactions occur together and share some of the same molecular species, system behavior can become complicated. A two- or three-dimensional network representation can help us visualize what is going on. Figure 9.5a shows four simple reactions involving eight molecular species and in which species A, C, E, and G each participates in two reactions. Figure 9.5b shows these same four reactions as a two-dimensional network. The two-headed arrows indicate that the direction of each reaction is determined by the concentrations of the reactants and products.

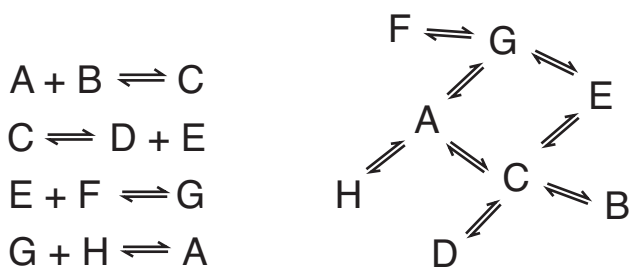


Figure 9.5. a. Four chemical reactions involving eight molecular species. b. A two-dimensional graph of the eight relationships implicit in the four reactions shown in a.

A quick look at figure 9.5b reveals that adding more of F to the reaction mixture will initially cause an increase in the concentration of G, which will then cause an increase in A and E, and so forth. By this means an increase in the concentration of F will cause a “flow” through the network ultimately increasing the concentrations of B, D, and H. If some F is added and some B is removed, one can visualize a more focused flow of material from  $F \rightarrow G \rightarrow (A \text{ and } E) \rightarrow C$  and finally to B. During the process the concentrations of every chemical species will change. The details of these changes are a property of the entire network and are not at all obvious if one looks only at the four reactions shown in figure 9.5a. When multiple chemical species interact in multiple ways, it is typical for stable states to form where the concentrations of some components change little even while others change dramatically, and it is possible for the concentrations of some species to oscillate or to exhibit even more complicated behaviors. Networks are particularly useful for visualizing material or information flows when multiple interactions are present in a system.

When the number of nodes and edges gets large, drawings of chemical networks become very cluttered; but even then, by selectively omitting some edges, a network perspective can significantly aid one’s ability to grasp the logic of chemical networks too complex to visualize in their entirety. Cellular metabolism provides a good example. A living cell maintains several thousand continuously interacting chemical species. The network is partly responsible for making a cell what it is and does. Over time, individual chemical species rise and fall in concentration depending on what has recently been eaten and the various

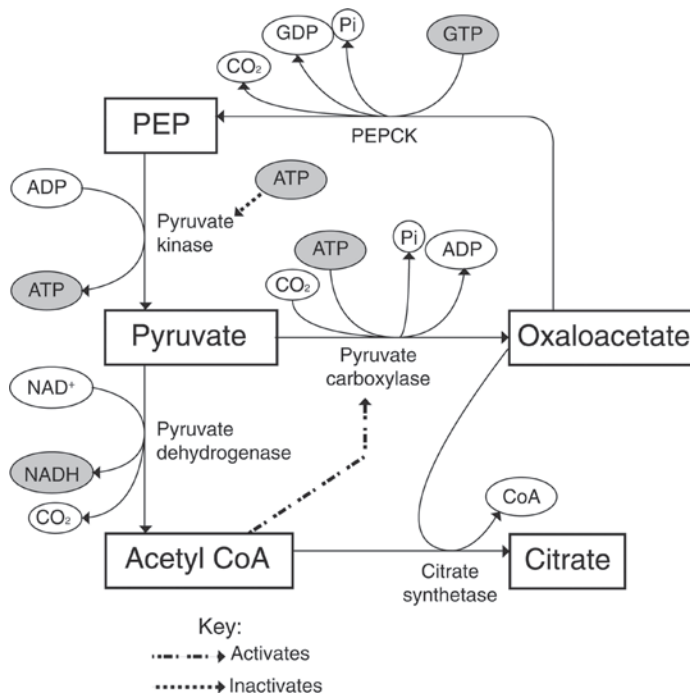
demands of cellular life. Overall, the network dynamically supplies a steady supply of chemicals needed for basic cell activities. A special feature of all cellular chemical networks is that every chemical reaction is both catalyzed and regulated by protein enzymes. Each enzyme is specific to a particular chemical reaction and controls the rate (but not the direction) of its reaction. Because of this feature, it is useful to label each edge (reaction arrow) of a biochemical network with the name of the enzyme that controls that reaction.

Since the activity of one particular enzyme is often regulated by concentrations of other specific chemicals in the network, it is also useful to include a second kind of edge, a “logical edge,” to network drawings of cellular biochemistry. Logical edges indicate control, information flow rather than a material flow. The information communicated by these links “decides” increases or decreases in material flow through material edges. Regulation is accomplished by changing the physical properties of individual enzymes. Figure 9.6 shows a small portion of a large biochemical network. This subnetwork involves twelve chemical species and has two logical edges. To reduce the spiderweb of edges, the chemicals ATP, ADP,  $\text{CO}_2$ , and  $\text{P}_i$  (a mixture of  $\text{HPO}_4^-$  and  $\text{H}_2\text{PO}_4^-$ ) are shown more than once. The entire cellular biochemical network of which this is just a small part includes over 2,000 chemical species.

Life-sustaining biochemical networks cannot exist without enzymes. Most chemical reactions that occur in cells will not happen at temperatures compatible with life unless a catalyst is present, and those few that do could not be regulated. Regulation of reaction rates is critically important to life. Dump the 2,000 or so cellular chemicals into a beaker, and very little will happen at room temperature. Heat the mixture, and the outcome would be a tarry mess, not a carefully orchestrated flow of chemicals supplying appropriate molecules when and where they are needed for cell growth and reproduction.

Organization and control result from the specific proteins that are present. We saw in chapter 4 how the structure and function of each protein is dictated by a sequence of nucleotides in DNA molecules. Information encoded in DNA determines what proteins are present in a cell and how much of each protein is produced at any particular time. Some of these proteins, in turn, determine what reactions occur and at what rates in the biochemical network. The overall result is a system that maintains the steady chemical environment that characterizes a cell.





*Figure 9.6.* A small aspect of a biochemical network that functions in a human liver cell. Black lines are material edges (chemical reactions), dashed lines are logical (regulatory) edges, chemical names are nodes, and enzyme names are edge labels. Abbreviations: PEP, phosphoenolpyruvate; PEPCK, phosphoenolpyruvate carboxykinase; CoA, coenzyme A; ATP, adenosine triphosphate; ADT, adenosine diphosphate; GTP, guanosine triphosphate; GDP, guanosine diphosphate; NAD<sup>+</sup>, nicotinamide adenine dinucleotide (oxidized form); NADH, nicotinamide adenine dinucleotide (reduced form); P<sub>i</sub>, inorganic phosphate; CO<sub>2</sub>, carbon dioxide.

## What is a genetic network, and what does it do?

The DNA in each of your cells directly encodes the information needed to make roughly 22,000 different proteins. About 10 percent of the specified polypeptides are enzymes, the rest have other jobs. DNA in the bacterium *Escherichia coli* (*E. coli*) encodes 4,322 proteins. There are organisms that have more genes than humans, and organisms with fewer than *E. coli*, but these numbers reasonably well bracket the raw

genetic material (instructions for proteins) needed to create and maintain a living organism on earth.

Even the casual observer must recognize that by any conceivable measure, a human being is much more than five times more complicated than a bacterium. *E. coli* is a single cell and adult humans are composed of an estimated 10 trillion ( $10^{13}$ ) cells. Human cells are characterized by dozens of complex subcellular structures including chromosomes, nucleus, mitochondria, a complicated vesicle transport system, and a dynamic (internal) cytoskeleton composed of millions of interacting proteins of more than 100 types. Bacterial cells have none of these structures. Human bodies have some 200 cell types organized into numerous tissues, organs, and organ systems including nervous and endocrine systems providing communication between and coordination of the different parts of the body, an immune system for fighting microorganisms that would use your body for food, a skeletal-muscular system permitting large-scale movements, and digestive, respiratory, circulatory, and excretory systems for managing the complex material needs of the body as a whole. The human brain has been called the most complex object in the universe (not counting the body, of which it is a part). *E. coli* has none of this. One of the big issues facing modern genomic science is to understand how the complexity manifest by the human body is possible with only five times as many genes as *E. coli*.

The answer, which is not yet understood in detail, seems to involve several things. First of all, multicellular plants and animals produce several modified versions of most proteins by a process known as alternative splicing. In this process several versions of messenger RNA are created from each (DNA-encoded) gene prior to protein synthesis. It is not known how important alternate splicing is in the creation of multicellular complexity, but this process is estimated to increase the number of distinct polypeptides produced by a human body to over 100,000.

Probably more important for understanding the difference between bacteria and people are the respective complexities of genetic regulation. A key feature of gene use in all cells including *E. coli* is control over what genes are expressed in which situations. In any particular human cell at any particular time, only a third to a half of the 22,000 genes are expressed, and when two cell types express the same protein, the amount of that protein is likely to be different. Differential

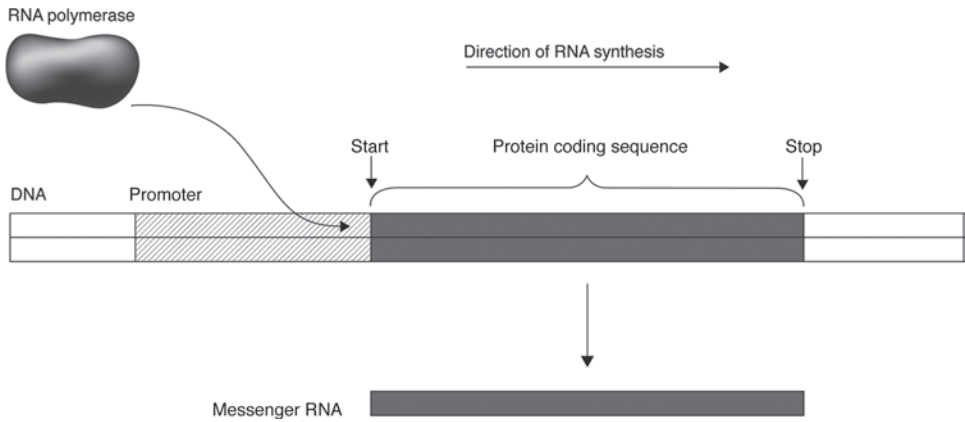
regulation of protein expression over time can generate different steady states with distinct properties in a network with thousands of nodes.

The unique complexity of every animal and plant can be traced back in time to a single cell, the fertilized egg or zygote. This starter cell is quite different from the hundreds of distinct cell types found in the adult body (with the exception of mature oocytes in the ovary). During formation of an adult body from a zygote, four types of things happen concurrently: the number of cells increases enormously by the mechanism of cell division, the physical characteristics of those cells change over time by a process known as cell differentiation, cells move and live or die in coordinated ways forming tissues and organs located in predetermined places, and the organism adds new material by taking in food molecules needed to sustain its biochemical network.

By means of these four activities, body complexity is built up sequentially through interaction of established structure with newly created cell types. Each interaction of cell type with structure leads to further elaboration of structure and more cell type diversity. Complexity builds on and amplifies previously established complexity.

Underlying all body-level changes are behaviors of individual cells with specific properties. These properties, in turn, are determined by the amounts and types of proteins that are present within them. Proteins determine the details of the biochemical network and also participate in the various structures present within each cell. Because the form and capabilities of the entire body result from the properties and behaviors of individual cells over time, ultimately we are what we are because of the differential expression and activity, over time, of those 22,000 genes in all the different cells. Genome analysis indicates that from three to five times as much human DNA is devoted to the regulation of gene expression than is used to directly code for proteins. In contrast, *E. coli* devotes less than 10 percent of its DNA to the regulation of gene expression.

It is apparent then, that to understand what we are (in terms of molecules), we need to understand regulation of gene expression. Figure 9.7 schematically illustrates the regulation of one gene. Regulation is achieved by controlling an enzyme, RNA polymerase, whose job in the cell is to make RNA copies of portions of DNA. As discussed in chapter 4, messenger RNA molecules are used by ribosomes to synthesize proteins. During protein synthesis, the order of the nucleotides



*Figure 9.7.* Typical gene structure with control sequences, called the promoter, located adjacent to a protein coding sequence. The messenger RNA is a copy of the DNA protein coding sequence.

that make up a messenger RNA molecule determines the order of the amino acids that constitute a protein molecule. Each messenger RNA directs the synthesis of many hundreds of identical proteins. Because messenger RNA molecules degrade, new messenger RNA molecules must be continually made if new copies of the protein specified by a particular gene are to be continually produced. Thus, primary control of protein expression occurs at the level of messenger RNA production.

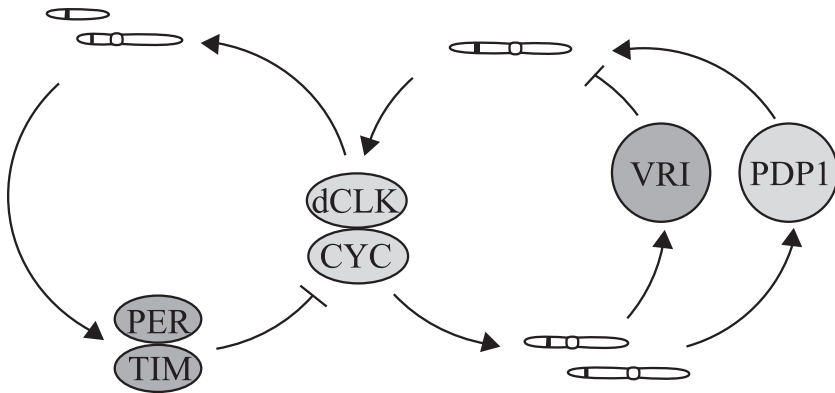
In figure 9.7 the double horizontal bar represents a DNA molecule that is thousands of times longer than the segment shown. RNA polymerase enzymes continually patrol the DNA in search of genes. “Start” and “stop” sequences indicate where an RNA polymerase is to begin copying DNA and where to stop. Copying always proceeds from left to right as the gene is drawn. A DNA sequence called the promoter determines whether or not an RNA polymerase successfully finds the adjacent start site. How frequently RNA polymerases make new starts determines the number of RNA molecules that get made, and this, in turn, determines how many protein molecules get made. Control of gene expression is achieved by determining whether or not RNA polymerases have access to start sites.

Proteins in the cell that bind to DNA promoter sequences are known as transcription factors. Because different promoter regions

exhibit different transcription factor binding sites, a unique combination of transcription factors bind to each promoter in a cell. Transcription factors may either block (repress) or enhance (activate) RNA polymerase action. The balance of the various repressors and activators binding at each promoter determines the activity of RNA polymerase on that gene. Since repressors and activators are themselves proteins, the amount of every repressor and every activator present in a cell is determined by other repressors and activators. It is a system of infinite regress. The amount of every transcription factor is determined by the concentrations of all other transcription factors present. Thus, gene regulation constitutes a network, much like a biochemical network, except that genes are the nodes rather than biochemicals.

This network has a logical structure that determines how much of each messenger RNA is produced. The network responds to changes in the cellular environment and makes adjustments in what proteins to synthesize and in what amounts. A major difference between bacteria (e.g., *E. coli*) and higher organisms (e.g., humans) is the number of transcription factors that influence the expression of each gene. For *E. coli*, the typical number is two. For human genes the number is often more than twenty. Because multicellular organisms (such as humans) have many more regulatory edges in their genetic networks, the complexity of their networks is much greater than for bacteria (*E. coli*). Presumably it is this difference in genetic network complexity that underlies and makes possible the much greater body complexity of humans compared to bacteria. A simple example of a genetic network is given in figure 9.8.<sup>2</sup>

As with electronic circuits and biochemical networks, the potential number of genetic regulatory networks is effectively limitless. Recent genome sequencing indicates that mice and humans have pretty much the same genes (nodes). The main difference is the structure of their respective genetic regulatory networks. Each network is what it is because of differences in promoter sequences and subtle structural differences in transcription factors that interact with those sequences. Since transcription factors and the DNA sequences they bind are all encoded in the DNA, all aspects of the genetic regulatory network are specified by instructions written as nucleotide sequences in DNA molecules. If there were no instructions, there would be no network and no organism.



*Figure 9.8.* The circadian clock, a genetic network. Circadian (twenty-four hour) cycles in the fruit fly are based on the cyclical expression of six proteins indicated by grey ovals. CYC and dCLK bind one another and activate expression of the genes located on chromosomes (elongated structures) encoding the proteins, PER, TIM, VRI, and PDP1. PER and TIM form a binary complex that inhibits the CYC/dCLK complex. VRI inhibits the synthesis of dCLK and CYC while PDP1 stimulates the synthesis of dCLK and CYC. The concentrations of the proteins produced by this network oscillate in a twenty-four hour cycle (see note 2).

## What is a proteome?

A full understanding of life requires inclusion of a third network, that of protein-protein interactions, sometimes called the protein-protein interactome, or proteome for short. At the molecular level, almost everything that happens in cells results from proteins binding specifically to something. This simple concept constitutes a fundamental aspect of modern biological understanding. Enzymes specifically bind to substrate molecules, gene regulation results from transcription factors binding to specific DNA sequences, RNA requires specific bound proteins to function, and proteins bind specifically to other proteins when building cellular structures.

Most proteins bind to other proteins. This network of interactions is the proteome. Some proteins interact with only a few proteins, while others interact with hundreds. Because enzyme activities are not just influenced by interactions with small molecules, but also by interactions with other proteins, enzymes constitute points of intersection

between the protein interactome and the metabolic network; and because transcription factors bind other proteins as well as DNA, the genetic regulatory network and the protein interactome are also interconnected. Thus, the three networks are best seen as aspects of a larger super network. Understanding the structure and dynamics of this super network is essential for understanding the cell; in fact, *it is the cell*. Because the super network is defined by proteins and because each protein is specified by a DNA sequence, properties of the network are ultimately determined by instructions encoded in DNA. DNA is an integral part of the super network because it is replicated and repaired by protein enzymes and because it provides the nodes of the genetic regulatory network.

### What is life?

All life, as we know it, is based on units of structure and activity called cells. The relationship between life and the cell is so tight it is fair to say that earthly life is defined by the little bundles of molecular activity we call cells. Some organisms consist of only a single cell while others, including ourselves, consist of large numbers of cooperating cells. The tripartite biochemical, genetic, and proteomic supernetwork that defines each cell explains how that cell responds to its environment, how it changes over time, and how it repairs itself, but it does not explain how a cell began. One more element is necessary for a comprehensive molecular understanding of life. This is the role of preexisting structure. In chapter 4 we saw how specifically designed proteins can self-assemble to form a bacteriophage. Cells create numerous structures by the mechanism of protein self-assembly. These include multienzyme complexes, ribosomes, and most structures of the cytoskeleton including various filaments and molecular motors responsible for movement. Like bacteriophages, these structures can self-assemble in a test tube when the appropriate proteins (and sometimes RNA) are mixed together under appropriate conditions. There are some cell structures, however, that cannot self-assemble *de novo*. These include DNA and most structures based on biological membranes including mitochondria and chloroplasts. It is possible that aspects of cytoskeleton structure also fall in this category.

All cells are bounded by a biological membrane and complex cells are divided into multiple compartments by other biological membranes. Membrane synthesis can be thought of as part of the biochemical network, and membrane-like structures can form artificially under laboratory conditions. In cells, however, new membrane is always created by adding new material to preexisting membrane.

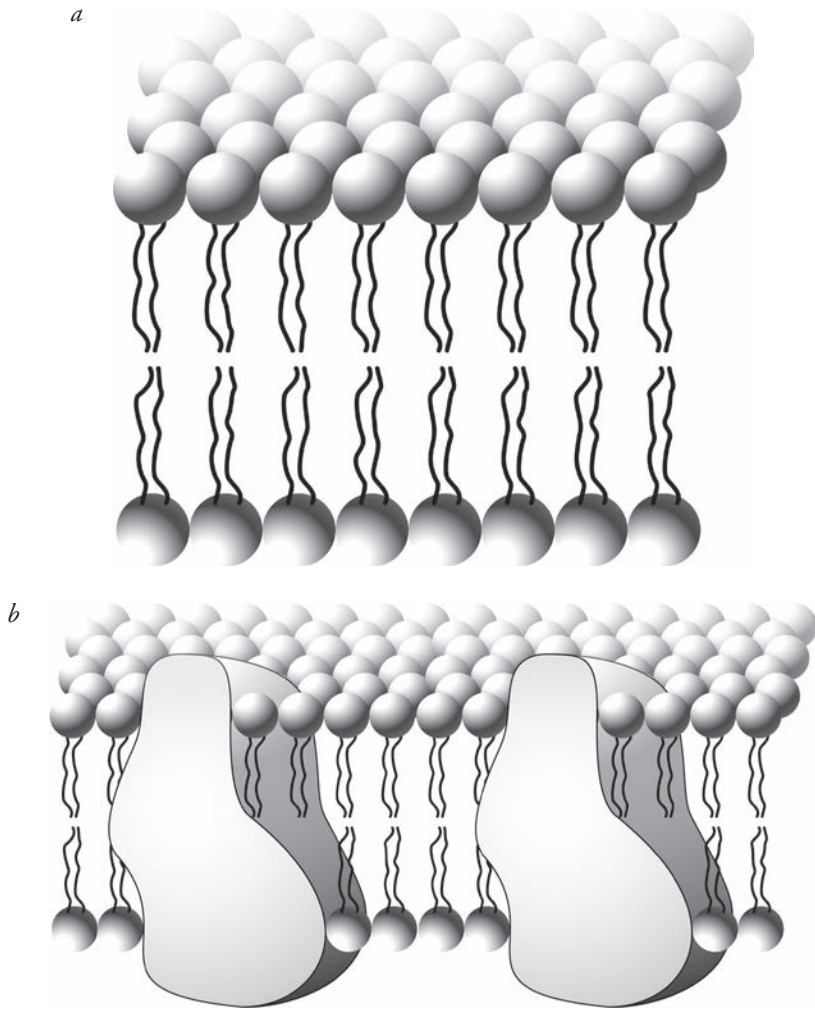
Biological membranes consist mostly of two classes of components, phospholipids and proteins. Phospholipids have the property that one end of each molecule is soluble in water while the other end is insoluble. A broad class of molecules has this general property, including soap, and most of these when mixed with water form an elongated ball, called a micelle, with the water-insoluble end of each molecule pointing inward away from water and the water-soluble end of each molecule facing out, making contact with water molecules that surround the structure. Phospholipids sometimes form micelles, but more often they form a two-dimensional sheet known as a phospholipid bilayer. A stylized view of a phospholipid bilayer is shown in figure 9.9a. In this figure, each phospholipid is represented by a ball-and-stick figure with the water-soluble end the ball, or “head,” and the water-insoluble end as two wiggly lines, or “tails.” The bilayer structure forms a thin sheet extending to the left, right, into, and out of the page in the figure. In a phospholipid bilayer, the heads are in contact with water, and the tails are in contact with other tails, thereby avoiding contact with water.

As a general rule, phospholipid bilayers are impermeable to water and water-soluble molecules but permeable to water-insoluble molecules. Since molecules insoluble in water cannot easily reach a bilayer immersed in water, phospholipid bilayers create barriers to the free movement of most chemicals.

All biological membranes have proteins imbedded in a phospholipid bilayer. These proteins modify the physical properties of the bilayer allowing selective passage of certain small molecules and transmission of environmental information across the membrane. The potato-like shapes shown in figure 9.9b represent membrane proteins. Actual proteins have more complex structures as illustrated in figures 4.3 and 7.3.

Figure 9.9 shows bilayer structures as flat sheets with exposed edges. This is not the most energetically favorable configuration, and such structures are rarely encountered in nature. When formed artificially, bilayers curl over so the free edges join together forming a ball, or





*Figure 9.9.* a. A phospholipid bilayer. b. Proteins in a phospholipid bilayer.

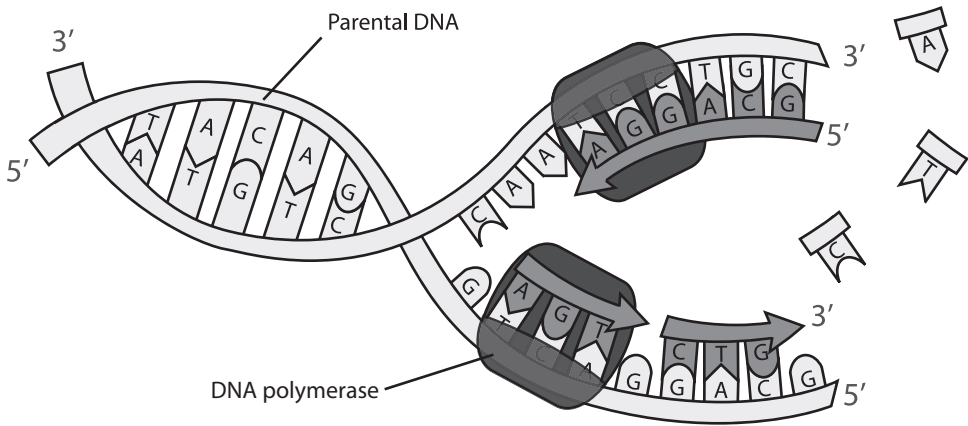
vesicle, with water trapped in the interior space. This arrangement eliminates exposed edges and isolates the inner water-filled chamber from the watery solution outside the structure. A cell is a large vesicle, and its cytoplasm is the trapped internal watery volume. Many cells also contain numerous smaller vesicles within them. Always, membranes determine what can enter and exit the inner volume. Every cell and every vesicle maintains an internal chemical environment different

from that outside of it. Regulation is accomplished by membrane proteins and their interactions with the tripartite supernetwork located interior to the membrane barrier.

The activities of membrane proteins are determined by their design, by interactions with other proteins, and by the concentrations of molecules that are constituents of the metabolic network. Because all parts of this system result from interactions among all the other parts, the supernetwork, in a sense, determines itself; but to do so it must always be encased in a biological membrane of its own making and regulation. This membrane keeps the molecular parts from floating away and allows the creation of a distinct local chemical environment. Cells make new membrane by synthesizing phospholipids and appropriate proteins and inserting them into preexisting membranes, but there seems to be no cellular capability for creating membrane where none previously existed.

This same property of only building on preexisting structure holds even more obviously for DNA. DNA molecules are always synthesized as copies of preexisting DNA molecules. The process is diagramed in figure 9.10. DNA in the cell is a double structure consisting of two very long molecules bound together by a type of weak chemical attraction known as a hydrogen bond. The two strands twist around each other into a configuration called a double helix. Within the double helix, the two strands are not independent structures. Whenever one strand has the base A (adenine), the other strand has a T (thymine). This is because A and T fit precisely together forming hydrogen bonds with one another. Likewise, whenever one strand has a G (guanine) the other has a C (cytosine), and G and C fit precisely together also forming hydrogen bonds. It is hydrogen bonds between A and T and G and C that hold the two long strands together in a double helix. If they don't match, they don't stick.

The exact complementarity of the nucleotide sequence of one strand with the nucleotide sequence of the other means that each strand carries the same information; if you know one, you can reconstruct the other. Complementarity provides the mechanism for accurate copying of DNA molecules. In figure 9.10, a small portion of a DNA double helix is illustrated. During replication the two strands are forced apart, exposing the nucleotide bases. DNA polymerase enzymes create new strands, each complementary to one of the existing strands. When the



*Figure 9.10.* Replication of DNA. The starting (parental) strand is to the left, it has the form of a double helix. To the right are “daughter” strands in the process of forming. The copying process moves from right to left. When finished, the two daughter strands will have exactly the same chemical structure as the parental strand.

process is finished, two DNA double helixes result, each identical to the parent molecule and containing half of it.

No cell has a capability for creating new DNA molecules without using this template mechanism. Creation of untemplated DNA sequences would be a disaster for the cell. Random generation of DNA sequences, of necessity, can result only in junk messages for the simple reason that the probability of creating something useful by random means is impossibly small.

There are other structures in the cell that can only form by growing or copying preexisting structure, but membranes and DNA make the principle clear. In high school biology we were taught that cells reproduce by division. Large cells separate into two smaller cells that subsequently grow until they are large enough to divide again. An essential feature of cell division is that copies or fragments of key structures end up in both daughter cells. If this does not happen, the cell cannot create them anew. Cells as a whole are never created by *de novo* assembly like bacteriophages. For this reason biologists have no choice but to envision a continuous chain of cell divisions leading back through history. All evidence indicates that such a path extends back from every

cell living today to a common ancestor cell, or more correctly to a single population of ancestor cells.

Geological and fossil evidence indicates that this population of ancestor cells lived more than 3.5 billion years ago. It is not necessary, in principle, that all current life be derived from one population of cells, but the observation that all known cells share many specific features strongly indicates the existence of a population of ancestor cells that possessed those same properties. Any other scenario is just too improbable to be believed. A short list of things common to all modern cells include: genes made of DNA, the same genetic code, messenger RNA, ribosomes, phospholipid-based membranes, numerous biochemical network features, and many proteins with similar structures and functions. The common ancestor was already very complicated, and so it could not itself have been the first life on earth.

The well-informed reader may have noticed that except for messenger RNA and tRNA we have not mentioned various other roles for RNA in the cell, and have not mentioned the other large class of molecules characteristic of life, polysaccharides. RNA and polysaccharides are both essential aspects of cells, but for different reasons they have been left out of our simple account. Polysaccharides can reasonably be considered an integral part of the biochemical network, while RNA molecules function by interacting with proteins or by acting like proteins, and except for messenger RNA, their functions can be formally included in the protein-protein interaction network without conceptual loss. Some may disagree with this assessment, but it would not significantly change our general view of cells as complex networks to add an RNA-only subnetwork to the supernetwork previously discussed.

What then is a cell? According to the view presented here, it is a tripartite network consisting of deeply entwined subnetworks: biochemical, protein-protein, and genetic regulatory. This supernetwork exists in a structural and chemical environment that it both propagates and maintains. The supernetwork has the remarkable property that it is self-instructed and self-maintained. The instructions for every aspect of this on-going structure, except directions for how to get it started, are encoded in the nucleotide sequence structure of DNA. Earth is characterized by a very large number (more than  $10^{20}$ ) of such networks; and each organism (i.e., network) can, in principle, be traced back to a population of networks (i.e., cells) that existed more than 3 billion years ago.

The nodes and edges of these supernetworks consist of DNA sequences, proteins, protein interactions with other components, chemicals, and chemical reactions. The synthesis and expression of the DNA sequences is controlled by proteins, as are the various chemical reactions. Even the various chemical species that make up the nodes of the biochemical network owe their presence in the cell to protein enzymes and membrane transport proteins. Proteins are always specified by DNA sequences that are in turn controlled, repaired, and replicated by proteins. The network is very plastic as attested by the heterogeneity of cell types in our bodies, and by the huge diversity of organisms and organism behaviors found in nature. All these properties derive ultimately from the dynamism of complex networks interacting with their environments.

A basic feature of networks that characterize life is that many, perhaps most, of the nodes and edges are not individually essential to the existence of the network. Whenever a new gene is expressed in the cell, new nodes and edges are added; when a gene is shut off, various nodes and edges disappear, but the network as a whole continues on with stability; it collapses only when the cell dies. As far as we know, the three subnetworks have persisted together since before the first common ancestor of all modern cells. In cells of multicellular organisms with multiple cell types, network properties in different cells can be significantly different reflecting that only a third to half of the genes in the organism's DNA are expressed in any particular cell. In a sense, each cell is a subnetwork of a larger body-wide network that is achieved by cooperation between the individual cells. Each cell network and the body-wide network traces back to a single cell, the fertilized egg, and before that to an unfertilized egg and sperm cell, and before that to cells in other bodies, on and on back in time to that hypothesized ultimate origin cell. This unbroken chain of historical causality is a fundamental aspect of life on earth.

### What is a complex adaptive system?

A living organism provides a prime example of a complex adaptive system (CAS). Other examples are a national economy, ant colonies, ecosystems, and the human brain. All are characterized by a degree

of stability when confronted with changing external events. Concepts that are important in the study of CAS are emergence, self-organization, adaptation, homeostasis, communication, and cooperation.<sup>3</sup> All CAS can be viewed and studied as networks, but not all networks are adaptive. What sets CAS apart from other complex systems are nodes with memory. Memory allows responses that take past events into account. In the study of these systems, nodes are frequently described as “agents.” Agents can be pretty simple, but not too simple. A sandpile exhibits complex behavior, but is not a CAS; neither is a solar system nor a thunderstorm a CAS. Sand grains, uninhabited planets, and atmospheric molecules are too simple to generate the range of adaptive behaviors characteristic of CAS.

Complex adaptive systems are characterized by nodes (agents) that make decisions based on past actions and on new input. All known CAS are characterized by nodes and edges that are prespecified, and most if not all can be described as a network with regulatory edges. Without careful prespecification, any particular combination of nodes and edges that behave as a CAS is simply too improbable to occur spontaneously. Proteins and their interactions in the life network, neurons in the brain, organisms in the biosphere, and traders in the stock market provide just a few examples of prespecified nodes with memory.

Extensive specification requires instructions, and because the engine of complexity is the only known way of generating complex instructions, ultimately, the engine makes CASs possible. Sometimes, the engine of complexity also operates within them. We will next turn to two systems that illustrate CASs very well, the human brain and human social institutions.

---

## Human Learning and Creativity

How are we to understand the human mind?

Picture yourself on a tropical beach with blue sky overhead, wind rustling in the palm trees, waves breaking, and white sand squishing between your toes. It's easy; anyone who has been to a beach can do it. This same capability allows mathematicians to prove theorems and sculptors to envision a finished statue in a block of marble. The mind is the greatest of all intellectual challenges to understand. It has been a subject of philosophical discourse for as long as there has been writing, and few great thinkers have failed to weigh in on the subject. By studying the brain, that physical organ in our heads, scientists are making progress toward understanding the mind.<sup>1</sup>

Biologists agree that the brain is a physical organ no less than the heart or the stomach. As such, its functions are not mystical even if we don't fully understand them. Every tissue and organ in the body performs certain functions, and all these functions are based on the coordinated activities of vast numbers of cells. There is little doubt that the central function of the brain is information processing. A lot has been learned

over the past fifty years about how this function is carried out, but there remains a question: how does an information-processing organ produce a conscious sense of self, the “mind”? Some find this hardware-centered perspective offensive or threatening, but the alternative is to indulge in mysticism that hinders clear understanding. To embrace a nonbiological explanation of consciousness is simply to yield to the age-old desire to separate humanity from our animal roots—in other words, to make us special. By firmly taking a scientific stand, I do not mean to belittle the issue. Exactly how the brain produces self-awareness is one of the great unsolved questions in science, but practicing neuroscientists have little doubt that the question will someday be answered in terms of physical laws and computational and mathematical principles and see little likelihood that any new physics will be needed.

Consciousness allows us to combine current sensory input with things remembered, and by so doing, create a continuous context. This, in turn, allows us to make decisions based on where we perceive ourselves to be in this context. Consciousness allows us to grapple with complex and abstract issues and to carry out internal debates concerning the virtues of one action or one interpretation over another. Consciousness also allows us to trial run various actions. To quote Karl Popper, consciousness “makes it possible, ultimately, for our hypotheses to die in our stead.”<sup>2</sup>

But consciousness is not our chapter’s main focus; rather we seek to understand learning and creativity. A fundamental issue of brain research is how we are able to learn new things. Given that our brains are able to encode past events and past conclusions about those events, how are we able to expand on what we have previously learned? We have seen that the engine of complexity is inherently creative and is a powerful strategy for finding improbable solutions to difficult problems, that is, problems whose solutions occur in a vast space of possible solutions, most of which are useless. The brain must solve such problems all the time. So, does it employ the engine of complexity?

## Where is the information?

Brains are made of neurons and other cells, and neurons communicate with other neurons by transmitting action potentials through synapses,



and also by secreting chemical signals into the surrounding milieu. From an information science perspective, diffusible chemical signals cannot account for much of the information content of thoughts because chemical patterns are simple, typically providing positional information in the form of gradients or by modifying local neuron behavior. The number of possible chemical patterns is much more limited than the number of possible synaptic patterns, so the amount of information they can encode must be much less. Thus, the focus must be on action potentials.

Formally, one action potential passed from one neuron to another conveys one bit of information. By this reckoning, a nerve bundle carrying thousands of action potentials per second is also transmitting thousands of bits of information per second. This seems reasonable enough, but the evidence is strongly against the idea that meaningful information in the brain is actually encoded this way.

Neuronal control of muscle contraction provides one example of how action potentials convey information. When the brain directs a body movement, the stream of action potentials continually impinging on every muscle cell in our bodies is modified. For some muscle cells, the frequency of action potentials increases, causing contraction of those muscle fibers, and for others it decreases, leading to relaxation. As a body moves, the brain continually modulates action potential flows to large numbers of muscle fibers in a way that results in coordinated contraction and relaxation of millions of muscle cells. Imagine the brain as a puppeteer controlling thousands of marionette strings. The flow of action potentials is like the tension on the string. The information directing muscle movement is not encoded in individual action potentials but rather in time-averaged increases and decreases in action potential flow.<sup>3</sup> Something like this probably happens in the brain too, although more complex patterns must play vital roles.

Throughout the brain there is a continual chatter of action potentials, no neuron is quiescent for very long. This flow is partly the result of neurons that spontaneously generate action potentials, but mostly it is stimulated by action potentials entering the brain from the senses and from action potentials arriving locally from other parts of the brain. Every action potential impinging on a neuron is communicated by means of stimulatory or inhibitory synapses. Receiving neurons act as microprocessors integrating all the inputs that arrive

every few milliseconds and responding by generating new action potentials whenever the time-averaged positive stimulation outweighs the negative over a brief time interval. Neuron-to-neuron connectivity patterns lead to never-ending flows of action potentials circulating in three-dimensional time-sensitive networks. Inhibitory synapses and anatomical structures shape the flows. Positive feedback loops provide for extended reverberations of particular patterns. At any moment an active network involves millions of neurons. Ephemeral networks rapidly recruit new neurons and then are replaced by new coalitions of communicating neurons.

Local networks function by interfacing with other networks. Timing is critical because action potentials arriving more than a few milliseconds apart show little cumulative effect. Synapses have adjustable strengths, and neurons have adjustable thresholds; so over longer time periods the impact of one neuron on another may be modified by strengthening or weakening the connections between them. Also, new synapses can be created and old ones lost. It is widely believed that learning involves the strengthening and weakening of synaptic connections reinforcing particular patterns. Strengthened patterns are those remembered. Stronger synapses make particular patterns easier to regenerate at a future time.

The details of three-dimensional temporal patterns of action potential flow that constitute an idea or thought or internal model must be incredibly complex, highly parallel, redundant, and noisy. Thoughts change over time and are probably different in detail every time they arise. It is extremely unlikely that the meaning of a thought could be deduced by summing up all the action potential bits that are propagating in a network at a particular time. Rather, the information must somehow be encoded in intensities, periodicities, or other general patterns of action potential flow. The idea is illustrated by simpler systems. For example, the information content of a message encoded in Morse code and transmitted through a wire cannot be measured by counting the electrons moving through the wire, but rather through the start-stop pattern of electron movement; likewise, the information content of a spoken message is not determined by the number of vibrations in air, but by patterns in those vibrations.

A major impediment to understanding the information-bearing structure of a thought is that the mode of information encoding in

the brain is very different from the orderly sequence of 0s and 1s that is the staple of computer theory or the linear flow of words that characterizes human communication. It is easy to see, taking a general perspective, that different connectivity patterns might represent different meanings. But, it is not at all easy to envision how three-dimensional cell-to-cell communication patterns can function in such a way that their outputs modify neighboring networks in just the right way to cause useful new networks to form. This is especially true of new networks that generate just the right sequence of muscular motions, or a proper visual image, or the outputting of a sequence of spoken sounds that conveys equivalent meaning. A related problem is the dearth of conceptual tools for comprehending exactly how three-dimensional networks consisting of neurons continually communicating with one another can encode a specific item of information. To put the problem in terms we have previously explored, how can a neuronal communication pattern act as an instruction? Neuroscientists cannot yet answer this question in specific detail.

To summarize, the information used to direct our muscles, or which underlie thoughts and mental images, or that represent chunks of learned knowledge, is imbedded in complex three-dimensional time-sensitive networks of action potential flows. This concept-level description is pretty vague, yet unless something really important has escaped the attention of neuroscientists, it more or less describes how the brain works. The model does not tell us exactly what a particular functional pattern might look like. Even so, if the model has any merit, we need to be able to explain in a general way how functional neural connectivity patterns form.

### How are decisions made?

Consider this: given that you know the first five bits of a six-bit string, say they are 10010, how can you predict the sixth bit? There are three possibilities. First, there may be a rule that determines the next bit. If you know the rule, then you can predict the next bit with 100 percent accuracy. Second, the next bit might be determined randomly, say by flipping a coin. If this is the case, then you can never do better than 50/50 at predicting the next bit. Third, the bit could be determined by

some combination of rule and randomness. In this case, if you know the rule, you can predict the next bit with greater than 50 percent accuracy, but less than 100 percent accuracy. Computational theory provides for no other possibilities.

For a person to “think up” the next digit is not a fourth computational option, but does suggest it might be interesting to examine closely what it means to “think up” a 0 or a 1. Perhaps we have a set of rules operating in our brains that in identical circumstances would always produce a 0 or a 1; or perhaps we are capable of mentally “flipping a coin” to decide such issues. And clearly, the brain might be capable of combining the two extremes in some way to produce an unpredictable, but biased outcome when presented with a recurring situation.

Psychologists exploring this issue face the reality that it is not possible to experimentally test which option actually happens in our heads. This is because it is not possible to arrange a human test that repeats *exactly* the same conditions. For people, the conditions of a test include whatever the test taker is thinking in addition to the task presented, and what the test taker is thinking might well include the previous trials in which he or she has participated. Everything that happens between the first trial and the second irrevocably changes the mental state of the test subject. So, exact repeats are not possible. We know that human subjects are not always predictable and are sometimes illogical, so it must not be that our minds function in completely deterministic ways. On the other hand, we are quite bad at generating random strings of 0s and 1s, so we probably can rule out the possibility that we have mental access to an unbiased random number generator. These observations indicate that we probably employ a combination strategy, that our mental processes are not random, but neither are they completely deterministic. This also makes evolutionary sense. It is likely that over evolutionary history a mixture of random action and predictability tended, on average, to produce the most successful outcomes. Rules that access memories allow us to take advantage of past experiences, while randomness introduces an element of unpredictability in our actions. Might we call this creativity? When faced with an adversary, unpredictability can provide a huge advantage. If, when pursued by a lion, wildebeests always turn left when the lion gets within ten yards and right when the distance is five yards, it wouldn't take very long for

a lion to anticipate each turn and simply meet its prey at the end of a series of evasive moves. Life is full of situations where unexpected (i.e., creative) solutions provide advantage.

When making a decision, we use information in the form of memories and also information received from our senses. Random fluctuations provide a third source of input. If we were to lie down in a completely dark, soundproof, and odorless room, letting our minds wander, unless we went right to sleep we would have no difficulty generating new ideas; and our thoughts would not be restricted to remembered events. Where does information characterizing novel thoughts come from? Introspection suggests that new thoughts come from new arrangements, new combinations of things remembered, but that they are not just a simple replay of old memories. The very notion of creativity suggests that this “playing around” with memory information is not strictly preprogrammed. It must involve an element of random experimentation. Creativity, after all, is by definition the generation of the unexpected.

The random bringing together of bits and pieces of remembered information, however, must inevitably produce a lot of scrambled thoughts, or mental garbage. These are thoughts that are of no use for the task at hand, or quite likely, no use for anything. How we focus in on useful thoughts must be important for understanding how we think.

### Could our world, and therefore our thoughts, be deterministic?

From Galileo’s time to the present, for four hundred years scientists methodically striving to understand how things work have established an overwhelming case for the notion that our universe is the result of an interplay over time of four things: discreteness (i.e., the countability inherent in an object-based universe), geometry, rules, and randomness. Much of our environment is discrete: individual objects exist. Because the universe is characterized by distinguishable things, counting and numbers are possible. If everything flowed smoothly into everything else, what would there be to count? Numbers would make no sense.

The rules come in two flavors: logic and physical law. Logic provides the basis of mathematics, and lucky for us, physical law is logical.

To my knowledge, no one has proposed a nonmystical explanation for why the universe is logical, or what an illogical universe would be like, or whether an illogical universe is even possible. Logic, when applied to systems of objects with specific properties in a geometric space over time, demands that regularities will occur and that these regularities can be described in mathematical terms. These are what we call physical laws.

The universe is also characterized by a great deal of randomness. A lot has been written about chaos theory, which is a collection of mathematical demonstrations that situations occur where far-in-the-future outcomes cannot be accurately calculated even though a completely specified deterministic mathematical description is known. Such situations are characterized by extreme sensitivity to key parameters that must be specified with infinite precision if calculations are to result in useful predictions at some arbitrarily distant time in the future.<sup>4</sup> An example is the “three-body problem” in astronomy in which three gravitationally attracting heavenly bodies mutually orbiting one another constitute a chaotic system. Calculation of future positions of the three bodies becomes less and less accurate the farther into the future one looks, even though the relevant laws of physics and the mathematics describing the relationships are well known and precise. The difficulty is that masses and velocities of three bodies cannot be determined with sufficient precision to allow perfect prediction of positions in space after many mutual orbitings have occurred. The uncertainty gets larger and larger the further into the future one attempts to predict. This kind of uncertainty is known as deterministic chaos; it characterizes numerous other systems including the motion of molecules in gasses and liquids.

In contrast to chaotic situations, which look random but can be described mathematically with great precision, the theory of quantum mechanics rests firmly on the postulate that sufficiently microscopic events are inherently unknowable except when expressed as probabilities. Two manifestations of this randomness are the fundamental unpredictability about when a particular radioactive atom will spontaneously decay and the impossibility of simultaneously determining the exact location and momentum of a very small object.

A particularly clear example was given by Arthur H. Compton, one of the founders of modern quantum mechanics. He put it in this way:

Now imagine a faint ray of light passing through a tiny hole, which then spreads by diffraction into a broad beam. In the path of this broad beam we may place two photoelectric cells, A and B, each connected to an amplifier. These will be made so sensitive that the entrance of a single photon, i.e., particle of light, into either cell is recorded. A shutter in the path of the light ray remains open long enough to transmit a single photon. Into which cell will the photon fall? There is no way into which we can be sure. . . . Though the first photon may enter one cell, with the initial conditions identical as far as any test can show, the next photon may enter the other cell. The result is thus not reproducible. It is as far as we can see, truly a matter of chance.<sup>5</sup>

Because the motion of an atom or molecule is altered when it interacts with a photon, every atom is a kind of photon detector. Whenever photons impinge on a gas or a liquid, the resulting changes in molecular motion are intrinsically unpredictable because there is no way to know which atom will be affected by a particular photon. When coupled with deterministic chaos, quantum mechanical uncertainty guarantees that our universe is pervaded with unpredictable-in-detail random atomic motion. This does not mean everything that happens is random—far from it—but it does mean that randomness is never very far removed and is always present at the molecular level.

When we think about an imaginary space of all things that are conceptually possible in our universe, the existence of random molecular motions implies that given enough time *any* local molecular configuration is possible somewhere, sometime. In contrast, if there were no randomness, then many situations would be permanently off limits. If our universe were deterministic, everything in the future would be determined at the beginning, and all actions, even our own thoughts, would be simply the playing out of a giant automaton. Probabilities in this situation would reflect ignorance but not uncertainty. In contrast, when randomness enters into a system, probabilities reflect more than ignorance, they also incorporate the unknowable. The upside of this is that a universe with randomness also permits creativity; in the absence of randomness, true creativity is not possible.<sup>6</sup>

Returning to the brain, experimental evidence strongly suggests that communication within the brain is “noisy.” Some neurons spontaneously fire. When nerves functioning together are probed experimentally,

individual cells are not behaving identically. Given that the specific activities of individual cells originate at the molecular level, molecular uncertainty must impact cell actions. Given all this uncertainty, it would be very surprising if the brain was a deterministic device. It is much more likely that the brain deals in probabilities.

### What are some general principles of brain function?

The adult human brain is estimated to contain 100 billion neurons, and each is estimated to average roughly ten thousand connections or synapses with other neurons. Thus, the brain has on the order of 1 quadrillion ( $10^{15}$ ) neuron-to-neuron connections. Synapses transmit action potentials directionally from one neuron to another. In chains of transmission, every neuron acts like a simple microprocessor creating a new action potential when stimulatory synaptic inputs dominate inhibitory inputs over a brief time interval. Some neurons also can initiate spontaneous action potentials. The “decision” about when to produce an action potential constitutes a simple (albeit chemical) computation. The transmission of action potentials with each neuron responding to stimuli from other neurons often follows a complex multipath that is determined by the geometry of the connections available and the collective microprocessor-like actions of all neurons involved. The result of vast numbers of simple single neuron computations taking place within an interconnected network having a particular geometry can be very complex and is definitely a computation.<sup>7</sup> Individual neurons can be grown in the laboratory, and when these are isolated from an animal brain it is possible to study simple communication patterns involving two, three, or some small number of cells; but within the human brain, circuits typically involve millions of neurons.

Brain architecture is modular, featuring groupings of neurons specializing in specific information-processing tasks. Local specialized computations are combined to accomplish larger tasks. One global task is the processing and integration of vast amounts of raw data generated every second by sensory cells. Our senses—vision, hearing, touch, taste, and smell—as well as an extensive internal monitoring system, depend on millions of sensory cells regularly reporting their current state by sending action potentials to other neurons that, in turn, are connected to



still other neurons in specific parts of the brain. A major function of the nervous system is filtering this enormous flow of raw data and identifying whatever might be useful either now or in the future and discarding the rest. This is a huge job, and one of which we are largely unconscious. Another major computational job performed by the brain is the orchestration of coordinated responses to the processed input data. This mostly means coordinating the activities of our muscles so that we step out of the way of an oncoming bus or catch a baseball with our hands and not our foreheads. Muscle control is accomplished by means of action potentials that travel down nerve tracts from the brain to individual muscle cells causing them to contract or relax. The brain is the regulatory center of a continuous, and massive, incoming and outgoing flow of input and output action potentials between the sensory cells and the muscle cells. The brain is also the seat of memory. Learning can involve conscious effort or can occur completely unconsciously.

When formulating a response to a pattern of sensory input data, it is extraordinarily useful to be able to correlate similar past patterns of input data with past outcomes. Even animals with very simple brains exhibit rudimentary memory. Animals also have desires. These may be as simple as “find food” or “try to mate,” or may be complex and multifaceted, as are interactions among people. Memories and desires influence the outcomes that brains generate. In addition, humans at least, are capable of creating an internal world model complete with inputs and outputs to and from other parts of the brain. We create mental images, visualize what’s on the back side of solid objects, “hear” sounds in our heads, and “feel” the results of imaginary actions. This internal model world allows us to explore possible courses of action without actually moving any muscles. Whenever the imaginary outcomes are judged to be desirable, real actions can be motivated.

Memory is necessarily a selective process; we don’t have the capacity to remember all the input data generated by our sensory cells, nor would it make any sense to do so, as most is of no consequence to our well-being. A big job carried out by the brain is to identify patterns both from the senses and from our internal model world that are “important.” When sufficiently important patterns are identified they are committed to memory. Memory implies storage of information. Brain design specifically makes possible the formation of vast numbers of states, each consisting of patterns of neuron-to-neuron communication.

Physically, any system that can take on multiple configurations is inherently capable of encoding information, and this physical principle must also underlie brain function. Because the brain is a physical network with active subnetworks representing aspects of the external world, we know that specialized information must be encoded in various states of the subnetworks.

Finite systems of discrete elements (i.e., systems with parts) are always characterized by a finite number of alternative configurations. Situations commonly occur where configurations of one system interact with configurations of other systems. In this way, one system can have varied but specific impacts on another system—like a driver (one system) interacting with an automobile (second system). Capacity for such interactions is what information measures. A brain, human or animal, is characterized by nerve cells (neurons) communicating with other neurons. The basic mechanism of communication is the action potential discussed in chapter 4. Each local communication pattern interacts with other communication patterns in other parts of the brain, and these interactions cause yet other patterns to be established. This general view is straightforward. The tricky part is understanding how specific subnetworks are created in such a way that they have specific meanings.

Analysis of the information content of three-dimensional systems in which the various components are physical objects whose interactions are governed by the laws of physics can become very complicated. Because of this, most information theoreticians focus on linear sequences of symbols that carry information. Strings of 0s and 1s serve the purpose very well, and fortunately, we can show that classical information encoded in more complicated systems can always be transformed by a suitable recoding into a string of 0s and 1s without loss of meaning. Because of this, computational principles that apply to the manipulation of 0s and 1s also apply to more complicated systems, including neuronal communication networks.

## What is evolutionary epistemology?

Epistemology is the formal branch of philosophy concerned with the origins and limits of knowledge. It is about how we know what we

know. In 1974 psychologist Donald Campbell coined the term *evolutionary epistemology* in a review of the works of philosopher Karl Popper on the subject of human learning.<sup>8</sup> The computational perspective provides additional insight. It brings to the fore the key observation that computers can copy the contents of another computer's memory, but we humans cannot copy another person's memories. We can learn from other people, but we cannot copy exactly what is in their heads. This raises the serious question of what is actually happening when we learn.

If you want your desktop computer to "know" something that is already "known" by another computer, you simply copy the appropriate file from one hard drive to another. The exact sequence of bits that encode the desired information is reproduced. When a person knows something, the relevant information is encoded in neural activity patterns in his or her brain. But, when two people know the same thing, their neural patterns are not the same. When we learn from a teacher, we do not copy the neural connectivity patterns in her head; instead we create our own, which, hopefully, conveys similar meaning. Likewise, when we read something, we do not encode the remembered information in our brains as a sequence of letters and spaces; rather we create neural connectivity patterns that represent the same content in the unique context of our unique brains.

So the central question about "knowing" something is how do our brains create unique neural connectivity patterns that convey the same meaning in one person's brain as does a different connectivity pattern in another person's brain? Answering this requires that we have a better idea of what a mental representation is. Mechanistically, what is new or different in our heads when we have an idea or after we have learned something new?

If you could shrink yourself down to the size of a cell and follow action potential transmissions from neuron to neuron, you would find that some paths would lead you back to your starting point, indicating the presence of feedback loops. Other paths would head off to distant parts of the brain where they interact with other feedback loops and with other data flows. Neurons in the brain are not randomly connected, but neither are the connections predetermined in every detail. The initial patterns are apparently determined by global rules, with many details left to chance; this is reviewed in chapter 7. Later when

we are born and begin to learn, connections are added and others are eliminated. The result is that every human brain is similar in overall design yet different in detail. This means that the exact same input to two different brains must result in different details in the paths and patterns of action potential transmission. This is quite different from what happens in two electronic computers, where the pattern of voltage and currents in two identical models is the same (within design specs), given the same input.

In a broad sense, the central activity of any brain is to “make sense” of an individual’s current state vis-à-vis the local environment and to formulate actions based on this understanding. “Making sense” means to form useful correlations between the current situation, memories of previous situations, generalizations of previous experiences, and possible future actions and consequences. Generalization is at the heart of learning. It is often the case that a series of events exhibits commonalities. It is much more efficient and useful to remember what is in common than to remember each experience as an uncorrelated entry in a list. The human brain is very good at finding patterns that relate one situation to another. These patterns when identified and remembered constitute conceptual understanding: meaning that can be communicated from one person to another, allowing different people to make similar sense of the same phenomenon.

Not uncommonly, input either sensory or conceptual, doesn’t instantly “make sense.” It doesn’t immediately register in a useful way; yet later it does. What happens in this interval? No one knows for sure, but certain things must be true. First, when we consciously know something, that knowledge must be a manifestation of patterns of neurons communicating with other neurons by means of action potentials. Because thoughts can be conscious for more than a fleeting instant, these activity patterns must have a certain stability that can only be achieved by signals cycling over and over in a pattern that involves feedback. Gerald Edelman, neuroscientist, prolific writer, and Nobel Prize winner, calls this “reentry,”<sup>9</sup> an ongoing dynamic interchange of signals between massively parallel neuronal structures with reciprocal connections.

Two other things we know are that connectivity patterns are dynamic and patterns involving one part of the brain have meaning to other parts of the brain. By this I mean that they must fit in, their

impact must be coherent and useful in a broader context. This is a lot to ask. Patterns of action potential communications propagating within localized regions must interface with other regional connectivity patterns creating coherent global patterns that add up to something meaningful. Every interface between regional networks inevitably stimulates new patterns of neuronal “talk” that interface with yet other ongoing patterns, and so forth. For a connectivity pattern to fit in (i.e., to make sense in a larger context) requires that whatever global behaviors (including thoughts) are generated, they must result in a useful outcome in the context of the current state of the organism, the outside environment, any desires in play, and relevant memories. This would seem to be a hopelessly complex set of demands, which brings us back to the central question of the chapter: how can connectivity patterns possibly get created that fit in meaningfully with all the other activities going on in the brain including memories and wants? In addition, we know that the brain does not contain a master engineer who designs a new connectivity pattern every time we encounter something new, or want something, or learn something, or have a new thought.

### What are some current theories?

In chapter 6 we considered artificial neural networks (ANNs). These are connectivity schemes that can be programmed in digital computers and that exhibit impressive abilities to learn. ANNs were inspired by neurobiology, and everyone who has studied the brain agrees that brain function is based on networks of neurons. Learning for an ANN involves adjustment of weightings that determine the relative impact of input at each node (see figure 6.1). In the brain weights are determined by the strength and nature of synapses. In computer models several methods are available for adjusting the weights. All involve training. In one method small random changes are made to weights, and then whenever changed values improve performance they are selected to serve as the basis of another round of modification and selection. When this strategy is employed, learning (a modified combination of weights yielding improved performance) results through cumulative selection—an evolutionary process. The larger the number of training examples used, the better the results. Another method is called

back propagation, and it is deterministic. This method requires that the desired output of the ANN and also the outputs of each node be expressible in numerical form. When this is the case, it is possible to write equations that permit calculation of changes to the value of each weight such that the difference between the present output and the desired output is minimized for a particular data set. Training involves the repeated application of back propagation with different data sets. The resulting weights are a compromise among the optimal values achievable for each individual training set and represent a generalization of the data sets used for training. Weight sets learned by either method usually will work well on new data sets to which the ANN has never before been exposed.

Several authors have analyzed the possibility that the brain uses back propagation to create appropriate connectivity patterns. All find the prospect highly unlikely. Neurons do have adjustable thresholds and adjustable synapse efficiencies, and it is quite likely that the strengthening or weakening of synaptic efficiency is an important aspect of learning. But, it is hard to imagine how the outputs of neural circuits can be expressed in numerical form; and it is equally difficult to imagine what anatomical structure could perform the needed numerical calculations and then implement the results by adjusting the individual strengths of millions of synapses. This would seem to require a second brain superimposed on the first and devoted entirely to the task of adjusting parameters. Brain anatomy does not exhibit such a parallel structure.

In recent years a new theory nicknamed the “Bayesian brain” has received increasing that attention.<sup>10</sup> In this theory, the brain is accepted as a network of neurons but a network that encodes probability distributions. A probability distribution associates likelihood with all possible outcomes. In this conception of the brain, perceptions are not what they seem. When we “see” a tree we are not directly accessing the sensory input from our eyes, but are instead perceiving our internal world model. Data input from our eyes is not perceived consciously, but rather is used by the brain to adjust our internal world model so that what we “see” in our mind’s eye closely reflects what is actually out there blowing in the wind. The match between what we see and what is actually present is never perfect, but the brain is continually adjusting its internal states in order to minimize the amount of mismatch between internally generated data (the mind’s eye) and sensory data

being generated by our sense organs. Statisticians call this minimizing the uncertainty. This theory grew out of studies of human visual perception, and visual data processing remains the dominant experimental domain.

The theory has several parts. First, is that higher brain centers develop and maintain mental models of different aspects of the world we live in and use these models to continually make predictions about what we have and are about to encounter. Second, is that predictions and sensory data are both encoded as probability distributions. This means that data and predictions both include estimates of their uncertainty. Third, is that a primary activity of sensory data processing is to minimize the uncertainty that occurs when incoming sensory data is matched with predictions. The mismatch (degree of uncertainty) between model predictions and incoming sensory data can be reduced in two principle ways, either by doing something to change the sensory data (i.e., moving) or by modifying the internal models. Modifying a model is a form of learning.

There are two reasons this theory has received so much recent attention. First, when experiments are designed to test how people respond in psychological tests, their responses turn out to be nearly “Bayes optimal.” This implies that we make decisions based on statistical inference of a type that was first described by Thomas Bayes in 1761 and which long after developed into a major subfield of statistics. Second, when the mathematics of Bayesian statistics is applied to networks that transmit data encoding statistical distributions, uncertainty takes the same mathematical form that free energy does in physics. This means that changes in brain patterns can be analyzed using the same mathematics that physicists use to analyze physical systems. Free energy is a thermodynamic parameter that physical systems minimize when they change. In a like manner, the Bayesian brain theory predicts that brain changes are driven by the minimization of uncertainty. Karl Friston calls this minimizing the “surprise.”<sup>11</sup>

In a neural network where each node has information about the uncertainty of its weights, minimizing uncertainty will maximize the accuracy of the output. If the output is a prediction and the uncertainty reflects mismatch between prediction and sensory input, then reducing uncertainty by adjusting the network weights amounts to improving the model. So, by minimizing the “surprise” the brain automatically

adjusts its internal models to more closely match internal predictions with sensory data.

This is an intriguing theory because it would seem to solve the problem of how the brain optimizes the weights (synapse strengths) in networks that encode models of our world. It also has a disturbingly deterministic flavor! In its simplest forms, the Bayesian brain concept seems to imply that the networks encoding the world model “automatically” adjust to optimally match sensory data. The Bayesian brain is a mathematical model that must be implemented in a physical device. Current attempts include some form of hierarchical brain organization in which higher brain levels project predictions made by internal models downward and lower levels project upward to higher brain levels the mismatch (uncertainty) calculated by comparing the projections with incoming sensory data. The uncertainty estimates lead to modified connection strengths that are the core of improved models that produce predictions that better match the incoming sensory data. This is an iterative process that over repeated cycles hones the internal models to produce predictions that well match the data. Experimental computer simulations are promising but run into difficulty when the models don’t closely match sensory data. There are at least two problems that can arise. First, the free-energy principle does not specify how to design a system that can build totally new internal models, and second, complex networks almost always have what are known in complexity science as “local minima.” A local minimum would be a situation in the brain where any change in the weights from the current best values would cause an increase in the uncertainty (an increase in the mismatch with the data), yet there exist other configurations farther afield in parameter and architecture space that would do a much better job if the brain could just find them. Because the Bayesian brain is a mathematical rather than a physical model, it assures us that it is theoretically possible for the brain to work, but it requires that a practical way to implement the theory that is compatible with everything known about the brain must be worked out. Progress is being made to accomplish this by building computer models, but the final story is not yet known.

There is another seemingly very different but well-known general strategy that is very good at solving open-ended problems and for which strategies are known for avoiding the local minima problem.



It is the engine of complexity. This is also an abstract concept that requires physical manifestation. Implemented in the brain, an engine of complexity strategy would need to keep alive multiple hypotheses (models) and have mechanisms for randomly mixing in elements of other further afield models that don't seem particularly relevant to the current problem and possibly even mechanisms for adding or deleting "random" network features. Most of these alternative trials would make poor predictions, but some of them might show promise and with further refinement might ultimately lead to much better solutions than the current best-candidate model. Such a strategy would require that during the brain processing associated with problem solving, the problem would need to be simultaneously worked on by multiple parallel attempts. Our analysis of the engine of complexity in chapter 5 showed that to implement the strategy efficiently, each round of refinement would need to generate multiple modified versions of each parallel thread being worked on, and each random or semirandom change would need to be modest in size. Everyone who has studied the brain comments on the massive parallelism of the cerebral cortex. The structure would seem ideal for carrying out the parallel processing required by engine of complexity-type computations.

Two attractive reasons for considering the possible role for the engine of complexity in the brain are: (1) that the approach has an impressive track record demonstrating that when cleverly employed, engine of complexity-type computations are quite capable of finding clever and previously unimagined solutions to hard problems, and (2) the scheme is compatible with Bayesian brain principles. When the engine of complexity concept is combined with Bayesian brain, one possibility immediately jumps out. Minimizing surprise could provide the criterion by which alternative hypotheses are selected. Another, but vaguer, idea is that the fast parameter adjustments (altering synapse strengths through feedback) that operate in simple implementations of the brain free-energy principle might dominate whenever the mismatch (surprise) between model predictions and incoming data is small. When, however, this fast process struggles and dissonance between input and model prediction remains stubbornly high, one could imagine that more wide-ranging attempts would be initiated that more obviously reflect an engine of complexity strategy. One can even

imagine this later process is what is happening when one is “thinking about” something.

These are exciting times in brain research. The Bayesian brain and free-energy principle have created a coherent framework for future study and are creating renewed excitement over the prospects for actually figuring out how the brain works.

## How do we invent?

In contrast to the uncertainty associated with how the brain learns at the level of neuronal circuits, there are numerous relevant observations at the level of conscious human experience. Dr. Ed Sobey, first director of the National Inventors Hall of Fame and founder of the Northwest Invention Center, travels the world promoting the idea that everyone can learn to be more creative. One of his techniques is to have groups of people invent something unexpected on the spot. All great inventors, he claims, invent by trial and error.<sup>12</sup> Thomas Edison was famous for this. It is what he meant when he famously said that “genius is 1 percent inspiration and 99 percent perspiration.” When confronted by a problem for which the solution is not obvious, inventive people try things out, test the better ideas, modify them, test them, modify them again, and test and re-modify again and again. It is not uncommon for the development of clever designs to involve hundreds of prototypes before a really good one is found. Many inventors say they “think with their hands.” What they mean is that when confronted with the task of making something to fulfill a need, they physically cobble together a prototype with materials at hand, manipulate it, modify it, try it out, and modify it again. This “hands on” process gives them ideas for trying again with improved materials and to modify the initial design even more. Sometimes the final product looks little or nothing like the first attempt. Dr. Sobey is quite clear that quick-and-dirty first attempts followed by successive rounds of improvement constitute a far more effective invention technique than prolonged pondering leading in one step to a finished product. Not explicitly stated in this description of the invention process is the much larger number of potential modifications that are considered and discarded in the inventor’s mind before each physical attempt is made.

The discipline of modern science works much the same way. The enterprise of science is an accumulating body of knowledge built up by cyclic hypothesis formation, testing, and hypothesis modification. Creative people are always throwing out new ideas for testing. But, thinking “outside the box” rarely, if ever, means coming up with something utterly new. “Brainstorming” is an established technique for getting a group of people to come up with as many ideas as possible, many of them screwy, in the hope of finding one idea that through improvement will lead to a better approach to a sticky problem. Societies are so much better at solving problems than are individuals precisely because a large number of people generate more varied potential solutions that can be honed by selection.

A fundamental insight of evolutionary epistemology is that the same principle must operate in our subconscious. When faced with a novel situation that requires new understanding, the best strategy, really the only strategy that could work, is to make some quick but nonrandom guesses based on whatever information is available (in memory) and then to modify and re-modify the best until a mental state is reached that does an adequate job. At the level of neurons sending action potentials to other neurons, this means the creation of neuron-to-neuron communication patterns that interface coherently with other neuron-to-neuron communication patterns and which culminate in successful actions.

### What is the evidence that the engine of complexity plays a major role in brain function?

The central message of this book is that a particular computational strategy, which I call the engine of complexity, is a remarkable method for efficiently building up sizeable and otherwise wildly improbable bodies of information that are purposefully useful, and that this method appears in various venues in our world. Human thoughts and human understandings clearly embody information. We know that thoughts are informational because they can be communicated using spoken and written language and because such communication streams can be analyzed and quantified in terms of their Shannon and algorithmic information content. We also know enough about the physical nature

of the brain to conclude that at the cellular level a thought is a complex pattern of neuron-to-neuron communications occurring in a system capable of generating vast numbers of other such patterns. Given that any one of these patterns has an impossibly low probability of occurring randomly, and given that the brain cannot “know” in advance how to construct a pattern fitting the next need, how does the brain create a pattern that solves the next problem?

The Bayesian brain hypothesis approaches this question from a statistical mathematical perspective and treats the brain as a statistical inference device. A probability distribution over the parameters of a model can be seen as encompassing “all possibilities” within the scope of that model. The free-energy principle provides an internally generated criterion for how the system can “home in” on an optimal solution. In other words it provides a means for selecting between different possibilities at the neuron level. These are important steps forward but still leave open the question of how the brain creates models very different from any previously generated. How do we deal with something we have never before encountered?

An appealing answer to this question is that new models are cobbled together by combining aspects of previously useful models. The system (i.e., the brain) could be designed to create new neural patterns by modifying and recombining old patterns. From an information science perspective, the information embodied in new patterns must come from somewhere, and the most obvious source is bits and pieces of older successful models. Still, this can't be the whole answer. An adult clearly has more information stored up in the brain than does a baby. Many experiments have shown that we start life with only rudimentary kinds of information hard wired in our brains: things like an ability to recognize a human face as special, an instinct for suckling, and a “talent” for mimicking spoken language. It is inconceivable that the complex thoughts of an adult person are nothing more than rearrangements of baby thought. There are only three possible ultimate sources for the information stored in our brains: it is inherited (comes from information stored in DNA), it is rearranged sensory data, or it is derived from cumulatively selecting randomly made small changes.

The fundamental problem faced by those attempting to understand human learning from a computational perspective is not that networks of communicating neurons can represent concepts and memories but

how *appropriate* patterns actually get made. Since the brain has a quadrillion connections, the number of unique networks made possible by its architecture is far greater than the number of atoms in the known universe. This establishes a capacity for encoding a lot of information. But, this does not solve the problem of how the brain creates the *right* network for any particular task. Only an infinitesimal fraction of possible networks can possibly interface in just the right way with other brain networks to produce an output appropriate for any particular input or output. The free-energy principle provides a compass for choosing one possibility over another, but it seems too much to ask that it also invent new architectures. Remember when thinking about this that the variety of potential inputs is essentially limitless. It includes such things as awareness of being watched by a tiger, desiring sex, listening to a lecture on calculus, smelling perfume, and innumerable others.

The underlying core difficulty is the probability problem we have discussed before in other contexts. Given that there are potential neuronal networks that would effectively integrate any particular input and appropriately interface with other brain networks to produce appropriately matched outputs, how does the brain find one that works well? The number of inappropriate networks would seem to be overwhelmingly large. It is easy to imagine, and experimental data supports the idea, that sensory data is screened, sorted, enhanced, and analyzed by hard-wired circuitry. But the resulting processed signals must then interface with other brain systems that weigh the filtered input in the light of emotions, memories, and conscious intents, then formulate courses of action, and sometimes but not always, carry out those actions. The activities are so varied and the demands of them so multifaceted that they cannot possibly be hard wired. New experiences demand new networks.

Nature has demonstrated in other contexts that engine of complexity-type computations are quite capable of solving open-ended problems as complex as those faced by the brain. If the brain does indeed implement some version of the engine of complexity, the implication is that there must be an unconscious effervescence of ever-changing combinations of thought and action fragments that provide the grist for selection of potentially functional patterns to be modified, tested, re-modified, retested, re-modified, and retested again until one matches

the job at hand. Out of uncountable numbers of possible action potential networks, one emerges by repeated matching and modification that fits the need at hand. In this process completely new aspects that have never before occurred may be part of the mix, but new aspects can only be a small part of each new attempt. It is probably no accident that this sounds a lot like the process of invention taught by Dr. Ed Sobey, but most brain activities are at an unconscious level. In order for such a process to work, as explained in chapter 5, each modification to the currently evolving information structure cannot be too large or else the system will bog down waiting too long for a too-improbable innovation. This would explain why we only learn effectively when we build closely on previous understandings and capabilities and why analogies play such a key role in comprehending something new.

Figure 10.1 shows the engine of complexity redrawn using vocabulary appropriate to brain action. The diagram illustrates abstractly how the engine of complexity would function in the brain cumulatively selecting networks that yield useful outcomes. The cycle starts with the generation of a modest number of “guesses,” tentative models in the Bayesian brain scheme (“outputs” in the figure) in response to a problem. Physically, these guesses would have the form of parallel neuronal communication patterns or networks. The guesses are tested by matching projected outcomes against the problem at hand. The poorest matches (those having the greatest surprise) would fade away. Those providing the best match would be probabilistically modified, resulting

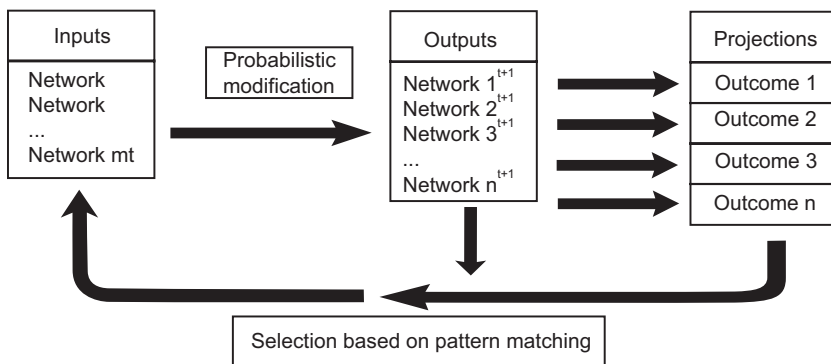


Figure 10.1. The engine of complexity reframed in terms of brain activity.

in multiple new outputs. These, in turn, are again tested against projected outcomes. Repeated cycles would rapidly result in better and better matches between the problem and projected outcomes. At some point a decision is made to “go with the best.” When there is great urgency, and the task is simple, a decision would need to be generated in a few hundred milliseconds. When the task is complex or the concept difficult, the process might take days or even months. How many people reflecting on their accomplishments have observed that when faced with solving a difficult problem, they needed to “sleep on it,” before the answer became clear?

I distinctly remember my own experience studying mathematics in college. More than once, when confronted with a new concept, I would read it over, work problems that illustrated it, and yet feel I didn’t really get it. A few weeks later, while working on something new, I would realize that the concept I struggled with earlier now made perfect sense; it was easy! What had happened? At the time, I had no idea what was going on in my brain but did think it very curious.

The engine of complexity method of learning becomes more visible when learning is a cooperative activity involving many people. Such activity is what generally falls under the umbrella of social evolution and includes all sorts of human activities characterized by the accumulation of cultural information over time, including things like transmission of science and technology concepts, economics, social structure, pop music, and religion. We will take a look at some of these in chapter 11.

To sum up, the evidence in favor of the engine of complexity operating in the brain is indirect and circumstantial, and, in the final analysis it is not known if the brain employs the engine of complexity to accomplish the remarkable things that it does. Selection is almost certainly utilized, but this is not enough to claim that the engine of complexity operates. For that there must be competition between parallel solutions simultaneously worked on and there must be a repeated process of selection, modification, and selection again so that the potential of cumulative selection is harnessed. Because the engine of complexity is such a powerful strategy and because nature is so creative at utilizing powerful solutions, it would be surprising if somehow this potential was not drawn upon, but only time will tell as neuroscientists steadily unravel how the brain works.

## What can we say about the origin of human creativity?

Viewing human thoughts and human technology as information-based phenomena and concluding that the engine of complexity must be responsible for assembling that information would take much of the mystery out of human creativity. It would seem to even resolve the age-old question of free will. After all, when the engine of complexity operates with random change providing the grist for selection, the outcomes are ultimately unpredictable. Cumulative selection solves the probability problem associated with the large amounts of purposeful information inherent to neuronal networks. Theoretical and experimental demonstrations indicate that the engine is most creative when random change provides the raw material for selection. Random addition and deletion to and from existing neuronal networks in the brain would establish the ultimate source for the information characterizing thoughts and provide a mechanism explaining why our thoughts and actions are not entirely predetermined. The alternatives would seem to be determinism in which everything we learn is ultimately a deterministic repackaging of sensory data and for which creativity is an illusion or random change *without* cumulative selection, which runs afoul of the probability problem.

The engine of complexity explains the creativity observed in biology and guarantees that the future of any evolutionary system cannot be predicted in detail. If implemented in the brain, it would also establish that human creativity was deeply grounded. The built-in creativity inherent in outputs generated by the engine of complexity would at the same time explain how we learn and why we are creative. Without its involvement we are faced with the seeming impossibility of creating neuronal communication patterns that can effectively represent any phenomenon or concept that we by chance might encounter without running afoul of the probability problem. One thing we can be certain of is that without some sort of random input, creativity is an illusion.



---

## Cultural Evolution

### What is human culture?

Human culture, as distinct from chimpanzee or ant culture, is made possible by human language. Beethoven symphonies, corner drug stores, presidential elections, the Roman Catholic Church, and baseball games all depend on language. Sophisticated language allows us to share complex ideas and knowledge with others. From a computational perspective, communication consists of transmitted and received information, and whenever a chunk of information is communicated widely among members of a society, we recognize it as being part of culture. A culture, then, is the outcome of many remembered communications. Animals communicate, but the amount of information they exchange is much more limited; as a result, animal “cultures” seem impoverished when compared to human cultures.

Historically, human culture has been utterly transformed at least twice, first by the biological evolution of language ability and second by the invention of writing. Both transformations enabled vast increases in cultural complexity. No one knows when humans acquired

sophisticated language ability, but the process must have taken millions of years to evolve because it was accompanied by changes in the anatomy of the larynx and huge expansions of several parts of the brain. Some students of the subject believe our modern abilities were only achieved within the last 100,000 years. Others speculate that complex language may have distinguished modern from Neanderthal man and provided our ancestors a decisive advantage. Also, human intelligence is tied to our ability to speak. Each of us knows a lot, and we learned most of what we know by using language, and much of what we think is framed in terms of the language we use.

Spoken language makes possible villages, traditions, crafts, and trade. The invention of writing greatly increased the amount of information that could remain current in a culture and also increased the number of people with access to it. This capacity is now being further enhanced by the digital revolution. Prior to writing, cultural information was limited to what could be remembered and communicated by individuals in a society. With writing, books and libraries could store information that no single person could remember in detail. The thoughts and accomplishments of thousands, even millions, of authors could be permanently remembered. With the Internet, such information is available in real time to anyone anywhere in the world.

Culture is not just about ephemeral things like fashions, pop music, democratic institutions, and religious practices, but also about physical artifacts, things like kitchenware, furniture, computers, and automobiles. All of these are made possible by the communication of complex information: instructions, recipes, laws, rules, and rules of thumb. These communications require language, and languages themselves are an aspect of culture that can be communicated and learned by others.

## How do cultures change?

The notion that cultures evolve is as old or older than the notion that plants and animals evolve. Many authors in the nineteenth century, including Auguste Comte and Herbert Spencer, wrote about the progression of cultures prior to the publication of Darwin's *Origin of Species* in 1859. Darwin himself mused about the "advancement of cultures" in the *Voyage of the Beagle* (1839). Darwin later expanded on these

thoughts in the *Descent of Man* (1871). But Darwin did not attempt to make an explicit link between cultural evolution and natural selection. This was left to later writers. Richard Dawkins coined the term *meme* in 1976 to focus attention on similarities between biological evolution and social evolution by hypothesizing units of culture, memes, that play a parallel role for culture that genes do for bodies.<sup>1</sup> The meme concept has stimulated much discussion ever since. Just for fun I typed *meme* into Google and got 376 million hits. This astounding number is testimony to widespread use of the term in Western culture. Since it was introduced, the term has acquired a new meaning, and many of those hits are because the word *meme* is now used by the Web savvy to mean an idea or concept that circulates on the Web.

Scientifically, two weaknesses of the meme concept are that it is difficult to think of an experiment that would disprove it and that it is not possible to define its physical structure with the same sort of precision we are accustomed to in genetics. A gene, after all, is a molecule (DNA) with a defined chemical structure that carries information in the order of nucleotide pairs (see chapter 4). A gene does a specific thing; it codes for a protein, or an RNA molecule, or serves as a binding site for a regulatory protein. The concept of a meme as a unit of culture is much vaguer. This has allowed critics to argue the idea is too slippery to serve as the basis for a serious scientific theory. It also means that the meme concept cannot be plugged directly into the established techniques used by population geneticists. Population genetics is a biological discipline specifically constructed to analyze the behavior of genes in populations of organisms over time. Attempts to create a mathematical theory of cultural transmission based on memes lack the rigor and precision expected by population biologists. Dawkins himself has never defended the meme concept as a rigorous scientific theory, but rather has described it as a useful rhetorical device.

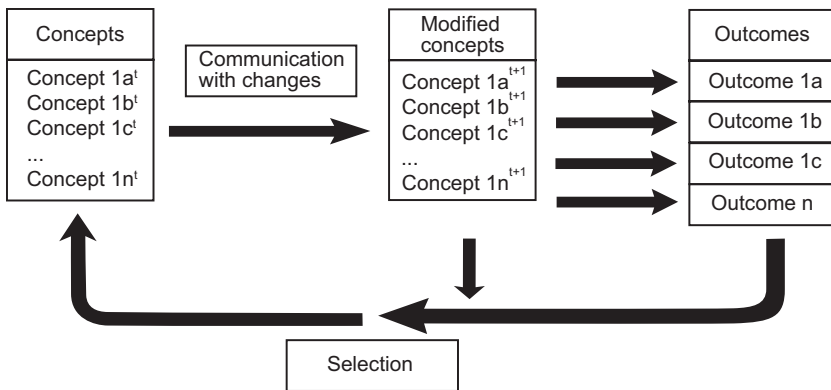
Fortunately, the more general notion that culture is based on shared information without trying to draw specific parallels between information use in cells and information use in societies can provide significant insight. One of the many virtues of defining evolution in computational terms is that it allows us to explore issues of cultural change even in the absence of a precise understanding of the physical structure of cultural information. The fundamental notion of a bit, the smallest unit of information, suffices. One such issue that can be tackled is the

long-standing question of whether or not cultural change is evolutionary in the same sense that biological change is evolutionary.

An obvious difference between cultural evolution and the evolution of organisms is the notion of reproduction. When an organism reproduces, information (in the form of DNA sequences) is transmitted from parent to offspring. Following transmission copied information resides within the chemical structure of a new individual. Likewise, each new antibody produced by the body requires a new generation of B-cells that have novel DNA sequences (chapter 7). The evolution of a thought in a human brain involves formation of a sequence of neuronal communication networks with each being a discrete physical pattern in the brain. With culture, the concept of *amplification* seems to better describe what happens than does reproduction, though one can imagine an idea being “reproduced” in another person’s mind when it is learned.<sup>2</sup> Regardless of what word is chosen, the communication of an idea from one person to another must involve some sort of copying.

Cultures are based on shared information, and key activities of individuals participating in cultures are transmission of information and selective remembering. Three requirements of engine-of-complexity-type computations (figure 5.2) are: a body of information that is copied, imperfect copying (i.e., copying with the possibility of modest change), and selection that determines which of the multiple copies will be copied again. This cycle is redrawn in figure 11.1, showing how easy it is to adapt the general definition of evolution presented in this book to human cultural communication. We saw in chapter 8 that there is no requirement for changes made during the copying process to be strictly random, though an element of randomness makes possible a wider range of future outcomes. The bottom line requirement is that there are outputs with differences providing a basis for selection to operate. If all outputs are the same, nothing changes over time. In an interesting variation on the theme, many rules of selection seem to be generated internally by the culture itself. Some examples of this are fashions in popular culture, markets selecting products, and the rise and fall of religious dogmas over time. So we have a situation where the rules of selection are often a product of the culture in which they operate. As such, most of the rules in play are subject to selection.

We are all familiar with the unreliability of verbal communication. The children’s party game “telephone” is entertaining precisely



*Figure 11.1.* A redrawing of figure 5.2, the engine of complexity, showing how the basic scheme fits person-to-person communication. In this figure inputs and outputs have been changed to “concepts,” and probabilistic copying has been changed to “communication with changes.” Various versions of the same concept labeled 1a, 1b, and so forth are circulating at the same time, and each time a version is communicated, with or without change, the superscript is incremented from  $t$  to  $t+1$  or from  $t+1$  to  $t+2$ , and so forth.

because of this unreliability. The game is made more entertaining when the players are careful not to repeat exactly what they were told, but becomes disappointing when players change the message entirely. It is most entertaining when individual changes are small. Written language provides stability to communicated information, but even with the written word, people find it hard not to “improve” a message before passing it on. Improvements are changes, and whether random or not, any change provides grist for selection.

But, culture is more than communicated information; it is information that has acquired a degree of stability through wide acceptance. Figure 11.1 does not explicitly show how individual people are involved. Generally speaking the information that characterizes a shared concept resides in people’s brains (or in a storage medium such as a book or computer hard drive). For information to contribute to culture there must be closed communication loops. When a concept is communicated from one to another, and then passed on to yet another and another, it eventually finds its way back to the original communicator in modified form. Only when this return communication is similar

to the original message is a sufficient degree of stability achieved for the information to have more than fleeting persistence in the culture. Because of its greater permanence, written information tends to achieve this state more easily.

To illustrate how the engine of complexity is integral to the process of cultural evolution, we will now explore three specific examples: scientific advancement, economic growth, and religion. I could have chosen many others, such as the evolution of languages, political institutions, technological advancements, or popular music—anything really that is a part of human culture—as all aspects of cultures evolve continuously except for those facets that are archived in a permanent medium such as books.

### How does science work?

Because of the pervasive influence of scientific understanding on modern society, a great deal has been written about how science produces useful information. The two most influential works on this subject are Karl Popper's *The Logic of Scientific Discovery* and Thomas Kuhn's *The Structure of Scientific Revolutions*. Of the two, Popper's work has had more influence on how scientists themselves think about what they do, and Kuhn seems to be more quoted by humanists.<sup>3</sup> Science is a complex cultural activity, and both authors have made important contributions to our understanding of it as an abstract subject.

In Popper's view, and in the view of most working scientists, science is fundamentally the application of a process that when carefully implemented solves a problem. The process begins when a working scientist asks a question: something about the natural world that he or she does not understand. Next, a hypothesis is formulated that if it were true would solve the problem posed by the question. These two steps are nothing unusual and are carried out hundreds of times every day by everybody in ordinary life situations. What sets science apart from many other human activities is that no hypothesis is of much interest (or value) unless it can be tested by experiment. It is easy to dream up explanations for happenings in the natural world, such as aliens living on the back side of the moon or spirits in the forest, but less easy to think up explanations that can be tested, and *experience has shown*

*that the great majority of hypotheses, no matter how appealing, are wrong.* This is a simple consequence of the computationally provable fact that the number of wrong explanations is always vastly larger than the number of right explanations. When something is believed without evidence, the odds of it being correct are miniscule. Only when hypotheses are tested against the actual behavior of nature do they gain credibility. The day-to-day activities of most scientists are spent designing and conducting experiments that test working hypotheses. This is why “scientific creationism” upsets biologists so much. It is not because religious explanations of our world might be right, but rather because they cannot be tested. In the absence of experimental testing, explanations are not scientific, and the history of nonscientific explanations of natural phenomena is not a pretty one. When they can be tested, they almost always turn out to be wrong.

A not-so-obvious aspect of science is that the purpose of experimentation is not to prove a hypothesis but to disprove it! An explanation that in principle cannot be disproved may be appealing but is ultimately vacuous. When the result of a carefully planned and executed experiment is consistent with a hypothesis, this does not prove the hypothesis but gives it credibility. When the outcome of a carefully conducted experiment contradicts a hypothesis, this means either the interpretation of the experiment is wrong or the hypothesis needs modification.

Another key feature of science is that experiments and their results must be published for others to read and repeat. Individual human logic is quite fallible. All of us fall in love with our ideas and unwittingly bias our interpretations, and scientists are just as fallible as anyone else. Only when others not closely associated with the hypothesis critically examine the design, execution, and interpretation of each experimental test does the validity of an experimental result become credible. Even then there can be hidden problems. When a hypothesis and its experimental support are published, and other experts believe it has merit, they devise new ways to test the idea. Only after a variety of experimental approaches produce results that are consistent with a hypothesis does it become widely accepted. Even then, no hypothesis in science is forever secure. One can never know if some new experiment will reveal a previously unnoticed inconsistency. Newton’s law of gravity was as firmly entrenched as any scientific theory for two hundred years until various experiments in the twentieth century showed

that Einstein's theory of general relativity provided a more complete account of gravitational phenomena than did Newton's law. The equation that Newton proposed turns out to be a special case of a more general way to explain nature. The special case works very well under conditions that prevail in places like Earth but does not describe at all well what happens in extremely high gravitational fields that occur elsewhere in the universe. There may be better, even more general, ways of understanding the attraction of masses to one another that have yet to be discovered. Time will tell.

A concept difficult for most beginning college science students to comprehend and accept is that science does not deal in ultimate truths but rather in "best explanations available." How can it be that science can serve as the basis of so many technological achievements when we don't know if accepted scientific laws and theories are true? The answer is that well-supported scientific theories are true enough to work well within a particular domain of situations even though the future may reveal more complete ways of explaining the same phenomena. Laws with limited domains work well in engineering; if you are building a bridge, it doesn't matter if Newton's law of gravity fails near a black hole.

Hypothesis generation, experimental testing, and hypothesis modification has proven to be an effective strategy for searching through the vast space of possible explanations to find one that fits what we observe. This process is described very well by the method diagrammed in figure 11.1 and thus satisfies the definition of the engine of complexity. In essence, the process creates a fit between knowledge and nature. The search process is not random, but on the other hand, no scientist ever knows whether or not her most recent hypothesis will hold up. Experimental testing provides an objective basis for the selection of ideas. Without such grounding, there is no way to know if a particular explanation is consistent with physical reality. When a hypothesis measures up, it is published and other scientists pick it up and expose it to more experimental tests, and if it still measures up, publish it again. Well-supported hypotheses get incorporated into textbooks and thus become more widely disseminated and tested. Dissemination and acceptance play the role in science that reproduction does in biology.

This evolutionary view of how science works does not in any way belittle the role of intelligent thought in the creation of hypotheses



and the design and interpretation of experiments but is rather a testament to how complicated our universe is. It is trite but true to say the universe is too complicated for anyone to understand in its entirety. A consequence of this is that no one can predict how aspects of the universe that are not yet understood will behave in the future. Scientific explanation allows us to make such predictions, for those aspects of nature that are well understood. From an information science perspective, the way one discovers something new is to search for it; and when the information space to be searched is too large to search in its entirety, and when you don't know exactly what you are looking for, the search engine diagramed in figures 5.2 and 11.1 is the most efficient way known for finding something that matches your goal. The method of science implements this technique in a human societal setting.<sup>4</sup>

Eric Beinhocker describes the role of human thought in the creation of cultural variation (generation of new hypotheses, etc.) "deductive tinkering."<sup>5</sup> When people come up with a hypothesis, whether it be a scientific theory or an explanation for why an automobile driver just behaved badly in traffic, it does not just come out of the blue but is based on things they already know. In science a new hypothesis (that has merit) must be consistent with and based on other more-established theories. Many a tentative theory has foundered by contradicting the laws of thermodynamics. These laws have been tested so many times and in so many ways it is very unlikely someone will find a violation in our corner of the universe. A characteristic of great scientists is they have a better grasp of what has been discovered and accepted by others than do run-of-the-mill scientists. This gives them an edge when designing new hypotheses because theirs' better accommodate all the information available; but even great scientists often guess wrong.

Accepted scientific theories and laws constitute a body of information that encompasses what we collectively know about the universe. Ultimately science is a social enterprise that collects, maintains, modifies, and adds to this body of information. The information is recorded in written form and resides in the collective brains of living individuals, scientific journal articles, reviews of selected literature, and text books; it was created through the repeated generation of variation (hypothesis formation) followed by selection (experimental testing and selective retention). When boiled down to its essentials, scientific knowledge is the output of a learning algorithm carried out by multiple scientists

posing questions, guessing answers, testing those guesses, modifying the guesses, and testing them again. This method is well described by figure 11.1. What sets science apart from many other human enterprises is an objective method for selecting which good ideas are consistent with nature and which are not. Over the course of hundreds of years, the accumulated knowledge that has resulted from this process is impressive and supports an ever expanding technological capability that gives humanity ever increasing control over our environment and our destiny.

### Economies change, but do they evolve?

In *The Origin of Wealth*,<sup>6</sup> Eric Beinhocker presents a strong case that economies thrive or falter through a process that involves the variation, amplification, and selection of business plans. Simply put, a business plan is a strategy for gaining value by making and trading something for something else you want. Business plans are instructions for commercial activities and enterprises. Business plans are often not written down, but they always could be. In traditional economic theory, trades do not happen unless both trading partners think they will gain value in the transaction. As such, trading is not a zero sum game: the more trades that are made, the more value is created.

Long ago all business plans were strategies that involved bartering, the exchange of one sort of thing for another with no set price. Following the invention and acceptance of money, most trades involved exchanging things for money. Because of this, things that are made, bought, and sold acquire a value that can be measured in money. In economic theory, value is established by what buyers are willing to pay and what sellers are willing to accept. When something is bought and sold a lot and the prices of these trades widely communicated, then its “value” is the rough average of many trades. Any business plan that has a seller producing and offering an item for more than anyone will pay falls into disuse, and any business plan that does not provide sufficient value for the seller cannot long exist because the seller will quit making the item. Many plans are devised and then selected against by the market. We could say they go extinct. Good business plans on the other hand generate value for both buyer and seller and tend to be

emulated by others. Emulation is the mechanism by which a business plan spreads (we might say reproduces) through a culture. By deductive tinkering, people are constantly making “improvements” to existing business plans, and the market positively selects those that have merit and discards those that don’t. When trades involve money, enterprises use money they receive to pay employees and buy raw materials; and employees and sellers of raw materials, in turn, use their received money to buy other things. In this way, money circulates. The more things that are bought and sold, the more money everyone has.

Business plans are not just about trading but also about the production of things. As plans are improved repeatedly through modification and testing, they accrue information. The resulting improved products and services have greater value because they can be produced more efficiently (increased productivity) and because they have more attributes that others want.

A central feature of Beinhocker’s presentation of wealth generation is that cooperation increases the efficiency of production. Modern economies are particularly good at generating wealth because they are particularly good at finding cooperative opportunities that produce products and services that do a better job of meeting people’s wants and needs at lower cost. This increased efficiency can be thought of as improved fit between the different parties’ goals. Always in a society there exist strategies that create value and strategies that don’t. A corporate business plan is a particular strategy for exploiting perceived opportunities. The Toyota Motor Corporation is the manifestation of a business plan that organizes the activities of a large number of people to produce automobiles of a quality and price that attracts people to buy them. Each buyer perceives his or her purchase as a “good deal” or they wouldn’t buy the car, and the company makes money at the price the cars are offered or it wouldn’t sell them. Increases in efficiency result in more units made that can be sold at a lower price. A few years ago when Toyota became the most successful corporation in the world at producing and selling cars, aspects of its business plan were already being copied by other car companies in order that they might become more successful themselves.

If you don’t believe that cooperative activities play a central role in the Toyota business plan, just contemplate what it would cost in money and time for you to assemble your own Toyota Camry from

parts that you purchase. Then contemplate the cost and effort required for you to fabricate all these parts from materials that could be found in nature. No individual could actually do this, but if it were possible, a person would need at their disposal the capability to mine and smelt various ores, a foundry for forging metal shapes, a machine shop for modifying those shapes, wire-making capability, computer chip fabrication capability, and facilities for forming many different kinds and shapes of plastics ranging from electrical insulation to vinyl seat covers to plastic bumpers. You would also need a chemical laboratory to supply the chemical feed stocks necessary to produce rubber for tires, paint for the exterior, and all the different plastics used. The cost of the resultant automobile is difficult to calculate both because the exercise is so silly and because no one who did not own a gold mine would have the money to purchase the needed tools and materials since they would not have time to do anything else to earn money. If everyone made their own car, no one could afford one. Adam Smith in *Wealth of Nations* (1776) made this point over two hundred years ago when he described the huge productivity increase achieved at a pin factory when pin making was divided into distinct operations and different workers were assigned to the different steps of the process.

The modern world economy is filled with vast numbers of situations where people doing specific jobs cooperate so that the collective outputs of their activities are products that can be offered at prices that are far less than if each worker made the same product in its entirety without the help of others. The lower prices made possible by the resultant increase in productivity lead to increased sales that, in turn, generate money that allows the people who perform the needed activities to be paid an amount they are willing to accept in return for their work. This pay, in turn, is used to purchase things that are affordable precisely because other groups of people are organized (according to business plans) in ways that permit other desired things to be produced cost effectively. A modern economy consists of large numbers of such organized activities interconnected in a complex trading network. The system is reminiscent of the cellular biochemical/proteome/generic regulatory network discussed in chapter 9 but with totally different kinds of nodes and edges. Underlying this network structure are business plans, instructions that specify the details of how each business (node) in the economy functions.

Beinhocker considers the economy to be a complex adaptive system (CAS) that incorporates many smaller complex adaptive systems. His definition of a CAS is a system in which large numbers of agents interact according to rules and in which agents have the ability to process information and adapt their individual behaviors in response to that information. In the economy agents are both individual people and organizations of people. Over the last thirty years, quite a bit of work has been done by scientists exploring the behaviors of CASs, and several features have emerged. First of all, the behavior of a CAS can be so complicated that its future is not predictable in detail. Such systems frequently exhibit emergent patterns and behaviors that are more than any simple adding up of the various inputs and constituent parts. Second, such systems tend to be characterized by large numbers of temporarily stable states that result from the interactions between agents in diverse parts of the system and inputs from outside the system. Third, a CAS is often able to transition easily from one stable state to another, and the sizes of observed changes are sometimes characterized by power law distributions. A simple example of power law behavior is provided by sandpile avalanches discussed in chapter 4.

When applied to business strategies operating in an economy, organizations that make money represent temporary stable states. The overall economy that is made up of the interactions of many smaller strategies also exhibits a degree of stability. Within this system there are always ways to make money, but always there are many more ways to not make money. It is inevitable that some businesses will be more successful than others. The really good ways are what Daniel Dennett calls “good tricks.”<sup>7</sup> The available good tricks depend on current details of the economy, and opportunities change over time. But always there are good trick strategies that when discovered and implemented will make money. The problem is that there is no way to know in advance which strategies constitute a good trick and which do not. Also, there is no way to know how long a good trick will last. The only way to negotiate this uncertain environment is to keep trying out new ideas and pursuing those that work, the process of deductive tinkering. Most often, successful ideas are modifications of older strategies.

Because what people want changes over time, and because technological advancements make new things possible, the details of the

economy are always changing, and successful business plans often do not last long. Any organization with a static business plan will soon find that another organization provides better value, and the unchanging business plan will lose out. You can't get any more evolutionary (in the engine-of-complexity sense) than that.

Two striking features of Western, and increasingly world, economies are the generation of unprecedented amounts of wealth and the unprecedented complexity of business plans. In Beinhocker's view these features are linked. As previously discussed, wealth generation is based on the increased efficiencies offered by cooperative behavior. The best strategies constitute good tricks that when implemented produce things people want at an attractive price. Beinhocker goes one step further and makes the claim that wealth is more than money; it represents a certain kind of order in the economic realm. He describes this as *fit order*<sup>8</sup>—that is, order that fits well into the forces and complexities presented by the economic network. In this way of thinking, wealth measures purposeful structure in the economy.

Why is it that business plans tend to get more and more complex? The answer has two parts. First, there are cooperative advantages to be found in ever larger organizations; these efficiencies occur despite the well-known fact that large human organizations tend to become bureaucratic, inefficient, and unresponsive to outside change. The free market plays the same role as the environment in biology: if inefficiencies outweigh the advantages of size, some other smaller business plan will outperform the larger one and force it to change or go extinct. A second reason for increasing complexity is that new innovations create opportunities that didn't exist before. In biology this is called "niche-construction." When the new opportunities are incorporated into business plans, they may become larger and more complex.

Thus evolutionary economic theory explains the seemingly inexorable increases we see in corporation size in terms of new opportunities and available good trick efficiencies made available by large size. No doubt the digital revolution and the corresponding increases in the efficiency of communication, data storage, and information analysis have opened up new possibilities that were not available even a few decades ago. Some corporations have been and will be better able to take advantage of these than others.

## Do religions evolve?

If a party of Alpha Centurions were to visit planet Earth and study humanity, there is little doubt that a chapter of their report would concern the importance of religion in most human societies. Perhaps they would find it curious that the details of these religions, especially the roles and powers of the deities, were rather different, yet the adherents to each religion were convinced that theirs was the only true belief. Another curious feature is that the deities never reveal themselves directly. For an ordinary person to communicate with their god, they must either speak through an intermediary (priest, priestess, or shaman) or have a faith that transcends any requirement for proof that messages from the supernatural are not mental aberrations. Many people believe fervently in the existence of superhuman existences that impact their lives on a daily basis, but whose very existence they cannot demonstrate to a nonbeliever. No wonder science and religion are often at odds.

Perhaps a telling feature of all religions in contrast to individual spirituality is that they involve human organizational structures. Every religion is organized with rules and dogmas, and human keepers of the rules and dogma. Religions also change over time. Population geneticist David Sloan Wilson persuasively outlines the process by which religions evolve in his book *Darwin's Cathedral*.<sup>9</sup> His central observations are that new religions are constantly popping up, that long-lasting religions provide tangible economic and social benefits in addition to intangible spiritual benefits to their participants, and that religions have long-term organizational structures that change over time. I would add to this list the observation that religions can be characterized in informational terms (this is true of all social institutions). The teachings, traditions, and organizational structure of every religion can be, and usually are written down. The function of this information is very similar to a corporate business plan.

Success or failure of any religion is determined by whether or not it attracts and retains believers. Religions that fail to attract and retain believers become historical footnotes. The pantheon of Roman gods went extinct because they lost out in the competition for believers with the Christian church, and the Peoples Temple (headed by the Reverend Jim Jones) ceased to function because the survivors of mass suicide abandoned their previous beliefs.

All religions promise things that cannot be verified. Primitive peoples live in a world filled with inexplicable happenings, and their religions tend to provide mechanisms (sacrifices, rituals, etc.) that protect them psychologically from the whims and angers of their capricious gods. Following the establishment of large societies made possible by agriculture and specialization of labor, successful religions promised hope beyond life. This may reflect the misery and hopelessness experienced by so many who lived in the difficult conditions created by these societies. The modern economy has lifted many out of abject poverty, and along with the improved conditions many Christian denominations tend more and more to emphasize satisfaction with life before death. People gravitate to a message that resonates with them and the conditions they face. In response, religions evolve—their business plans change. Those that do not adapt go extinct.

Wilson notes that successful religions tend to provide their believers with improved living conditions. All religions are social structures that provide means for organizing cooperative behavior among their participants. It is often the case that this cooperation provides tangible economic and social benefits to its participants. Population biologists have written reams discussing how altruism in animals can arise through natural selection. The difficulty is that self-interested individuals have a clear advantage when dealing with a do-gooder. This advantage disappears when the cooperators agree to collectively punish the selfish. Human cooperation requires trust and effective communication among the cooperators. Successful religions include behavioral rules that if followed enable cooperative behavior. By insisting that all participants adhere to the rules allows believers to trust individuals they have never met. This in turn enables large-scale cooperative behavior, and within the universe of cooperative behaviors are various good tricks that provide tangible benefits to the participants. Successful religions almost always have explicit means of enforcing “good” behavior. The payoff is that by participating in a cooperative community, conditions of life improve. Spiritual promises are easy for new religions to make, but the economic benefits of cooperation within a community provide strong incentives for belonging to a group of like-believing individuals. All successful long-term religions tap into this fundamental reality of human social organization.



Wilson argues that religions evolve on two levels. First, they compete with one another for believers. If given the opportunity, people “shop around” for a religion to participate in. So, religions need to present themselves in appealing ways. Religions that lose out at this level of selection go out of business for lack of interest. Since new religions are constantly being formed, there is a never-ending supply of choices available. The only time this is not true is when one religion gains a local monopoly and institutes rules making it socially difficult or dangerous to opt out. The second area of evolutionary change is the rules, dogmas, and structure that define what a religion is and how it operates. The constant need to appeal to parishioners means that the details of every religion are always being tinkered with; new features are continually being tried out. At this level, the evolution of religions is no different from the evolution of corporations. Both compete with other organizations for customers, and both are characterized by an evolving information-based business plan.

Business plans, whether for corporations or churches, evolve through the actions of individuals within the organization making small additions or deletions. These changes are then tested by selective mechanisms internal to the organization and by selection among organizations. Through this process, every aspect of a religion, even its deepest held dogmas, can morph over long periods of time into something quite different. These changes can be dramatic, such as when Jesus Christ appeared in Judaism or when Joseph Smith founded Mormonism, or quite gradually as illustrated by the changes that transformed the Congregational Church from the rigid moralism of the Pilgrims who settled Massachusetts to the tolerance espoused by its direct descendent, the United Church of Christ. In each of these cases, the informational content of an existing way of thinking and behaving was modified and tested in the marketplace of potential believers.

### How does the engine of complexity operate in human society?

The engine of complexity works on a body of information that is evaluated in some way, requires a mechanism for copying and modifying this information, and operates in an environment that provides consistent

selection favoring some, but not all, of the modifications. Society as a whole and most social institutions examined separately exhibit all these features. Controversies arise over the nature of the information and the mode of copying, but it is not all that mysterious. Fundamentally, the information that underlies and enables social institutions resides in people's heads. Neuronal information patterns that constitute memories are augmented by printed and digitally encoded information, but it all traces back to things people have heard or read and remember. Because of this, understanding of social evolution is tied closely to understanding how we encode and remember concepts and thoughts. The mode of copying is different from what we have seen in previous chapters. The copying needed for social evolution to occur involves communication of information from person to person. The information can be spoken, written, or in two- or three-dimensional visual formats, but unless it is remembered, it has not been copied. Placing a book in a library is not enough to enter some new insight into the fabric of a society, people have to read it and remember it.

Modification of remembered information is no problem. Human memory is so fallible and people are such storytellers that it is typical for each remembered thought to be modified every time it is communicated to another. Probably the biggest difficulty facing those who want to understand how societies evolve is not how change occurs but how new concepts retain enough similarity to recirculate in a stable form.

An important feature that must be present for the engine of complexity to operate at the societal level is consistent selection. Selection criteria may change, but if they change too rapidly, cumulative selection won't add up to anything. Societies are characterized by norms and a sense of how things should be. Cognitive scientists tell us that our minds attempt to impose order on the world through the construction of narratives and mental models. When these interpretations become widely shared across populations of minds, they establish persisting worldviews that provide a consistent basis for the acceptance or rejection of new ideas when they are presented. If an idea doesn't fit into this internally generated rule set, it usually doesn't persist.

This brief review should make it pretty clear that social institutions from corporations to religions evolve by means of the engine of complexity no less than oak trees, bacteria, immune responses, and

thoughts. All of these are ultimately rooted in the same computational strategy. We turn next to the frequently made observation that evolutionary (that is, engine-of-complexity-based) systems often seem to generate ever increasing complexity. We saw one example of this in the growing complexity of corporations in the modern economy. Does the engine have some hidden drive that biases its outputs to become ever more complex, or does every evolutionary system respond to unique system specific pressures?

---

## The Evolution of Complexity

### Can complexity be defined?

The perception that evolutionary change produces ever-increasing complexity is deeply ingrained in Western culture. There are some valid reasons for this, but scientific opinions about whether or not it is true differ, and the matter remains unresolved. On one side are incredible advances in science and technology and also the broad sweep of life beginning from simple chemical mixtures that eventually yielded self-aware human beings. These examples suggest the action of some inexorable drive that generates more and more complexity. On the other side of the argument are observations that many biological lineages do not generate any obvious complexity increases over long periods of time and direct experimental demonstration that evolutionary algorithms can be designed to produce either increasing complexity or decreasing complexity (discussed in chapter 6). Two examples of biological lineages that have apparently changed little over hundreds of millions of years are cockroaches and horseshoe crabs. Parasitic lineages characteristically show decreases in morphological complexity over evolutionary time.

These and other examples employ the same engine-of-complexity computational strategy, but have different physical forms, occur in different environments, and respond to different selection rules. Apparently, details matter. A large body of work on evolutionary algorithms provides convincing evidence that the engine of complexity when operating in a computer does not drive complexity one way or the other; rather, it is system details that determine whatever long-term trends we observe. Two clearly important details are the nature of variation generated and the specific rules that govern selection. The engine of complexity chooses among whatever variety is present in each generation, and when the scope of choices is curtailed, future outcomes become correspondingly limited. In the extreme, if no changes are ever introduced, the engine will endlessly maintain the status quo. When changes are taken from a finite set of possibilities, the engine will explore all of them and settle in on the best combinations available. When, however, change is structured in such a way that there are no limits to long-term change, the exploration can go on indefinitely. Small random changes provide a very effective way to accomplish this.

The rules of selection are also important. Evolutionary computer algorithms are often written that calculate some property of outputs and select the next generation of entities according to the best scores generated. Such algorithms can create outputs with increasing or decreasing complexity over time depending on what is measured and selected. In short, complexity can be selected for or against.

Selection in biology is based on reproductive success and does not involve calculating anything. Reproductive success depends on internal factors, interactions with the environment, and chance. The challenge for an evolutionary biologist trying to understand the origin of biological complexity is to find convincing links among organism features, the environment, and complexity measures. If complexity provides advantage, surely evolution will favor it, but why should complexity confer advantage? A similar problem faces students of social change. What advantage, if any, do more complex social institutions have over simpler ones? To the extent they are understood at all, the answers seem to be different for different aspects of culture. Thus, languages do not exhibit inexorable increases in complexity while in recent times at least, technology and the economy have. A completely different situation is presented by antibody production. When we have an infection,

antibodies are produced that exhibit better and better fits to the pathogen. Complexity does not seem to be the right way to characterize this; binding affinity and selectivity provide more useful and appropriate measures.

In this discussion I have been using the term *complexity* in a non-technical, common sense sort of way. Most of us are comfortable declaring that one thing is more complicated than another. But, to further explore the relationship between evolution and complexity, we need to better define what we mean by the term. If we think something has increased, we can only test the assertion by measuring it. It is obvious that a human being is more complex than an amoeba and that an iMac is more complex than an abacus, and it is not difficult to discover measures that document these claims; but how should we compare the complexity of an iMac to the complexity of an amoeba, or the complexity of a community of microorganisms with a community of humans? The problem is not that measures of complexity have not been devised. Lots of them have, but none seem to capture all aspects of what we intuitively recognize in something complex. The meaning of complexity is itself complex.

Many physical scientists view negentropy to be the proper measure of complexity. Negentropy is the difference between the entropy of a system and the entropy of the same system at equilibrium (i.e., when it has its maximum possible entropy). Since entropy measures the amount of disorder, negentropy measures order. An interesting take on this approach is to equate the negentropy of a body of information to the “amount of information” it carries about something else.<sup>1</sup> The problem with all negentropy-based approaches is that while order and complexity may seem similar, they are not the same thing. Two systems that differ in apparent complexity can have the same negentropy. Shannon information and entropy are mathematically equivalent, and, as we saw in chapters 1 and 2, this measure does not capture important aspects of complex objects. This observation is not changed by subtracting the actual entropy of a system from its maximal entropy.

In keeping with the general idea that something complex must have lots of parts and lots of interactions between and among parts, some people have found it useful to distinguish “structural complexity” from “functional complexity.”<sup>2</sup> This is an idea illustrated by contrasting the automobile one drives with a large box containing all the parts of the

same automobile disassembled. Both are characterized by the same structural complexity, that is, by the same kinds and numbers of parts, but the arrangements and interactions differ between and among those parts. Clearly, the assembled machine is more complex in some ways than a box of the same parts. For one thing, the functioning automobile exhibits behaviors while the box of parts does not.

Concepts encompassed by the notion of structural complexity include the number of parts, numbers of different kinds of parts, and structural scale. Scale is important. I disassembled a ballpoint pen at my desk and found eight obvious parts including the ink, but at a different scale the pen also contained vast numbers of atoms. One can ask: is it useful to consider the positions of all the atoms when assessing the complexity of a pen? If one atom is removed from a pen does it matter? As a general rule, when defining structural complexity of a macroscopic object, the scale of interest must be specified or we become overwhelmed by irrelevant atomic and subatomic detail.

Functional complexity encompasses behaviors, control of behavior, and time scale. Once again, scale is important. Some things happen too fast or too slowly to be noticed by humans. These often can be ignored with no ill effect. It all depends on what we are interested in. Francis Heylighen makes the important point that complexity is always assessed relative to the observer.<sup>3</sup> This is because every observer has a natural scale and because only certain aspects of a situation are important in any given context.

A different way to approach complexity is to consider its opposite, simplicity. Simple things have few parts, or perhaps many copies of one or two parts, like 1s and 0s in a binary string. When we compare objects with the same number of parts, simple ones tend to be repetitive and complex ones less so. Elements of structure can be repeated in one, two, and three dimensions like the atoms in a crystal and also over time like repeating patterns of sound. Also, repetition can be simple or complex.

Physicists analyze repetition in terms of symmetries. A very simple kind of symmetry is mirror symmetry; this occurs whenever one half of an object looks like a reflection of the other half. To describe a mirror symmetric object, one need only describe one half of it and specify the other half to be the same as a reflection. The number of possible symmetries is limitless. A few simple ones are two-fold translational (like a

copy made on a copy machine), three-fold rotational (a triangle), and repetition over time (a wave). Symmetries are important for understanding complexity because the more symmetrical an object, the less description it requires. An object with a six-fold symmetry requires less specification than one with two-fold symmetry because only a sixth of it needs to be described in detail. An oscillator repeats the same pattern over and over, but to describe this behavior the basic repeat need be specified only once. Symmetries also can be superimposed; thus, an object might simultaneously exhibit both two-fold and five-fold symmetries. Crystals exhibit translational symmetries and also rotational symmetries. From this perspective, the most complex objects are those that exhibit the least amount of symmetry.

Taken to the extreme, this line of thinking implies that objects with randomly arranged parts, such as a gas, are the most complex. But, a gas exhibits no structure, and structure is normally associated with complexity. A way around this awkward situation is to recognize probabilistic symmetries. Any part of a gas is statistically the same as any other, so when statistically averaged properties are considered, a gas is highly symmetric and therefore simple. To sum up this line of thought, a complex object is one with complicated symmetries. I do not find this to be a very helpful definition.

Computer theory offers other ways of analyzing complexity. In computer theory, linear strings of 0s and 1s are objects, and objects don't just arise spontaneously, they are the result of a computation. One advantage of the computational perspective is that digital inputs and outputs have an obvious natural scale. The collective behaviors of electrons that underlie the symbolic 0s and 1s manipulated in electronic computers, and the ink atoms that constitute 0s and 1s printed on paper are clearly irrelevant to the meaning of a computation. It's the 1s and 0s that matter, not what they are made of. Another conceptual advantage of digital computations is that the inputs and outputs are made up of only two kinds of simple parts, 0s and 1s, and this makes such systems easier to think about than systems composed of many kinds of complex parts.

What then are some ways computer scientists characterize the complexity of binary objects? We have already seen some. The simplest is to count the number of 0s and 1s. Shannon information, introduced in chapter 1, is based on probabilities and formally measures ensembles of objects but can be used to characterize individual objects if one



assumes the object belongs to an ensemble in which all the members have the same probability. Input to a computation also can be thought of as a description of the output. From this perspective, the length of the shortest input that results in the formation of an object measures its minimal description. The complete description of an object must somehow encompass its complexity; so, the length of a minimal description must be one characterization of its complexity. This is the idea behind algorithmic information, also called Kolmogorov complexity, introduced in chapter 2. Algorithmic information is greatest for objects that have no internal relationships, that is, those with randomly arranged parts. This is a disappointing conclusion from the perspective of our quest for measures of that which humans judge to be complex.

An aspect of object complexity not captured by length, Shannon information, or algorithmic information is the difficulty of computing an object from short input. Complex objects are characterized by many parts and many relationships between parts; so if a complex object is computed from short, unstructured input, the parts and their relationships must get created during the computation. An attractive idea is that the most complex objects are those that require the most computation to create.

This naturally raises the question: what exactly do we mean by “amount of computation”? In computer theory, time and space are considered resources. Space is required for memory (room to store bits), and time is required to manipulate data (transition between various states of a computer) and to store and retrieve bits from memory. Often time and space (i.e., memory) can be traded off. This means that if memory is restricted, the computation can still be performed, but will require more time. Conversely, long computations frequently can be shortened by making greater use of memory. Reading from memory takes time, however; so even if an infinite amount of memory space is available, the computation of a complex object may still take a long time.

The short input requirement for this tentative definition of complexity assures that details of the output are not simply listed in the input. Informally, in computer science output structure that takes a lot of time to compute from short input is called “deep,” while structures that can be quickly computed from short input are called “shallow.” Depth reflects the complexity of internal relationships that exist within

an object. The more complicated the relationships, the longer it takes to compute an output that contains them from input that does not. It is in this specific sense that depth provides one particular measure of object complexity.<sup>4</sup>

A fundamental contribution of computer theory is formal mathematical proof that objects exist that can be imagined, yet cannot be computed from input shorter than themselves. There are two distinct reasons for this. One is that the required computation might be so impossibly long that even though a short algorithm can be written for producing a particular long object, the universe will not last long enough for this result to appear. The other is that when considering all possible objects, there are more long ones than short ones. This is easy to show. Only two objects are one bit long (0 and 1), four objects two bits long (00, 01, 10, and 11), eight objects three bits long, sixteen four bit objects, and so forth. The number of strings of a particular length grows exponentially ( $2^1$ ,  $2^2$ ,  $2^3$ ,  $2^4$ , etc.) as length increases. In this sequence of numbers, the last term is larger than the sum of all the previous terms. So, there are always more potential long outputs than possible shorter inputs, and it must be that most long objects can only be computed from inputs at least equally long.

To compute any particular thing we must start with appropriate input. The question is: how can one find an input that yields a desired output when the shortest input that accomplishes this task is long? Formally, in computer science, to find something we must conduct a search for it. It is possible to design computer programs that search potential inputs for ones that yield desired outputs. If such a search starts with very short inputs and proceeds systematically to longer and longer ones, the number of inputs to be tested grows exponentially with the input length. This means that as the length of the inputs being tested gets longer, the search slows down. If the search attempts to systematically examine long inputs, the required search times quickly exceed the lifetime of the universe. This is bad news and means that long inputs cannot be systematically explored; but another even more fundamental problem surfaces with comprehensive searches. It turns out that some inputs result in computations that never end; the computer simply cranks away forever. Still other inputs result in very long computations that eventually do yield an output. Unfortunately, there is no way of knowing whether or not an arbitrary ongoing computation

will eventually stop producing an important result, or will simply continue computing away forever. This is known as the *halting problem* and means that no strategy exists for conducting an exhaustive search of long inputs in order to find all those that produce some predefined output even if the searcher should somehow have limitless time available to conduct the search.

What all this boils down to is that for some problems there exists a best answer but no way to find it. This may sound bizarre, but it is provably true. Luckily, the situation is not as bad as it sounds because there are always suboptimal inputs and resulting outputs that a searcher may be able to find and which will work pretty well for the task at hand. This is why it is possible to write computer programs that work.

A major problem with using depth in its typical formulations (and also algorithmic information introduced in chapter 2) to characterize objects is that most attempts to measure these properties run afoul of the halting problem. Except for very simple situations, one never knows for certain whether the shortest input or the shortest computation that outputs something has been found. In most situations these measures can only be approximated. In spite of this the concepts of depth and algorithmic information are very useful for understanding complexity.

## How is complexity defined by biologists?

In biology the term *complexity* is used in various ways. Sometimes it refers to the variety of life forms present on Earth. In this usage, as the number and diversity of life forms have increased over time, life has become more complex. This usage is common in ecology, where the complexity of an ecosystem is some function of the number of species and the interactions between and among them. A different meaning focuses on the complexity of individual organisms. Is a human being more complex than a fish? How would one document such a claim?

It is evident to anyone with a passing knowledge of the history of life that complexity increases have occurred over time. This is well documented for complexity of the first kind—diversity. Figure 12.1 shows the irregular, but statistically meaningful increase over time in the number of marine multicellular taxonomic families that have been described from fossils.<sup>5</sup> Increases of the second kind, individual

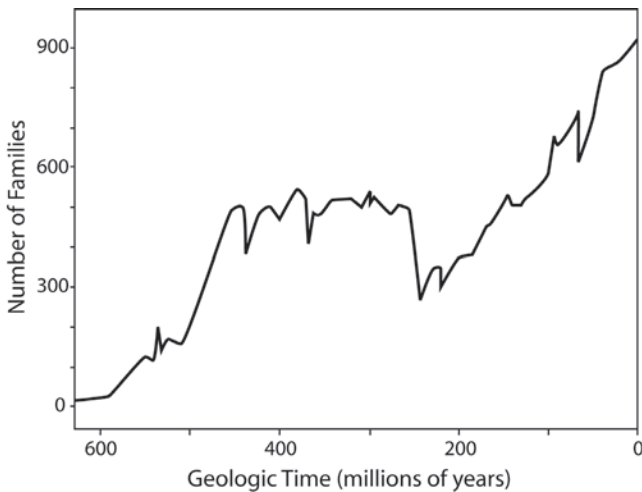


Figure 12.1. A plot of the number of recognized marine metazoic taxonomic families against geologic time. Long ago (far left) the number of families was small. At present (far right) the number of families is the highest ever (see note 5).

complexity, are also documented. After all there is little doubt that a human being is more complex than a bacterium.

Controversy arises when poorly defined conceptions of individual complexity become linked with notions of “advancement” or “progress.” Until the twentieth century, it was pretty standard in Western culture to accept without question that humans were the highest achievement of a progressive universe. This line of thinking dates back at least to the Aristotelian *scala naturae*, “ladder of nature,” an ordering of all things arranged with rocks at the bottom and people near the top, just below the godhead. The idea was incorporated into church dogma in medieval Europe with angels, archangels, and Christian God positioned successively above human beings. The concept continues to have powerful sway on Western thought. Some acknowledge Darwinian evolution and then incorrectly infer that the evolution of life is a march up Aristotle’s ladder. What nonbiologists often forget is that modern bacteria have been evolving just as long as modern humans, and when time is measured in generations, bacteria have evolved far longer than we have. By some measures, such as total numbers, total mass, and ability to cooperatively exploit diverse environments,

bacteria are collectively more accomplished than we. We are bigger and smarter, but who is to say that bigness and smartness are the best ways to judge advancement?<sup>6</sup>

Following publication of *On the Origin of Species* in 1859, Darwin's theory quickly became entwined with notions of progress and the inference made by some that evolution is goal oriented. That goal, of course, was seen to be humanity, especially northern European humanity. As a general rule, attempts to judge evolutionary progress, or advancement, get hopelessly entangled in anthropocentrism—a human-centered view of the universe. It is simply very difficult for us to evaluate other creatures and not conclude that they are in some way less advanced than we. This often unconscious bias finds its way into ranking systems. As a consequence, after appropriate “scientific study,” humans come out at the top of the ranking—exactly where we knew we were before conducting the study! Value-neutral measures are not easy to find.

Thermodynamics and/or information theory would seem to offer the best hope for objective measures of evolution-generated complexity. Length measures amount when the information containing structure is linear, but it does not discount repetitiveness. Shannon information and negentropy provide possible measures of accumulated non-randomness, but they do not account for the difficulty of computing something. Algorithmic information discounts repetitiveness, but is maximal for randomly configured objects. This hardly seems to provide what we are looking for. Depth, the difficulty of computing something from simple input, offers yet another alternative, but for non-digital objects we don't have any idea how to calculate it. Other measures have been suggested, but given the current state of theory, computer science does not provide a practical computable measure of organism complexity. This means that discussions of complexity in biology tend to lack a firm theoretical foundation.

In spite of this, it is pretty obvious that a human being is more complicated, no matter how we measure, than a single bacterium. This is easy to document. We are characterized by more parts and more interactions between and among parts. The fossil record, despite its imperfections, is quite clear that long ago there were only single-celled organisms; later, after eons of time, seaweeds and strange plant-like animals appeared. Still later, worm-like animals made their appearance in the seas. Even later, there were primitive fish. Only relatively recently

in the record did plants and animals invade the land and become even more complicated.

The fossil records of both land plants and land vertebrate animals document a steady appearance of forms ever better adapted to life out of the sea. Better adaptation is marked by various innovations and by organisms that dealt in more sophisticated ways with environmental challenges presented by life in dry environments. The earliest plants, some 450 million years ago, were something like modern liverworts and mosses that even today require rain to transport motile sperm to unfertilized egg cells. Later, there were large seedless plants similar to modern club mosses (lycophytes), horsetails (*Equisetum*), and ferns. These, in turn, were largely overgrown and replaced by seed plants, the ancestors of modern conifers. Later still, beginning about 120 million years ago, primitive seed plants were marginalized by flowering plants. The flowering plants are characterized by complicated strategies for pollination and by reproductive tissues packaged in a specialized structure—the plant ovary that when mature provides superior mechanisms for seed dispersal. Two examples are the fluff that transports dandelion seeds on the wind and fleshy tissues that induce us (and other animals) to eat grapes and plums along with their seeds and thus disperse their digestion-resistant seeds.

In the grand sweep of this well-documented history, each major “advance” is characterized by new plant features. These include stomata (tiny openings) in the leaves allowing control of evaporative water loss, vascular systems allowing plants to grow tall, and enclosure of the female reproductive tissues in an ovule allowing the efficient storage of food stores to support the growing plant embryo away from the mother plant. Later appearing features include enclosure of the ovule in a plant ovary allowing for better mechanisms of seed dispersal, and equally dramatic changes in male reproductive tissues that in ferns and other primitive land plants are small free-living plants (gametophytes) that produce swimming sperm. In flowering plants, the male gametophyte phase of the life cycle, also called pollen, is non-photosynthetic and so small that it can blow in the wind. In many ways and by every measure a modern flowering plant is more complex than a liverwort.

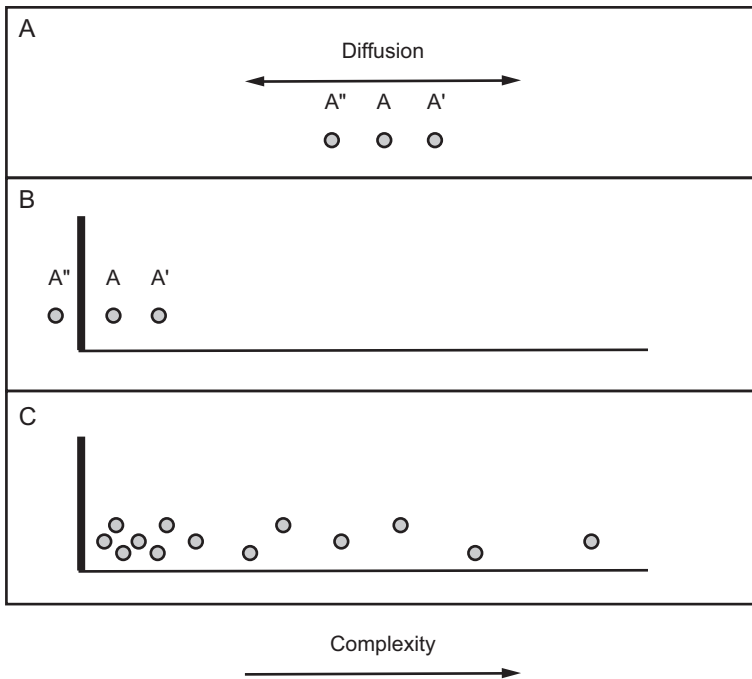
The fossil record of the vertebrate animals tells an equally dramatic story, featuring the familiar progression from fish to amphibians, to reptiles, to mammals, with birds and mammals appearing most recently.

Living mammals illustrate a progression from egg-laying creatures (monotremes), to those that give birth to embryos (marsupials), to placental mammals that nurture and protect the embryo and fetus inside the mother's body. In the primate lineage, the earliest fossil forms were little shrew-like creatures that lived 55 million years ago. These were followed by lemur-like tree dwellers, which were followed, in turn, by monkeys, apes, and in the last 7 million years, hominins—our immediate ancestors. The oldest fully modern-looking human skeletal remains are only about two hundred thousand years old. In the context of the 3.5 billion year history of life, this is incredibly recent.

It is hard not to see these evolutionary sequences as evidence for steady evolutionary advancement—progress of some sort. But, not explained by claims of inexorable progress is the continuing existence of apparently primitive forms. Thus, bacteria are still with us, so are seaweeds and amphibians, and also tree shrews, lemurs, monkeys, and apes. The exact species documented by fossils are gone, but as often as not some modern species seem not very different from their ancient ancestors. Of course, we don't get to examine the ancient creatures in minute detail to determine if they were less complicated at molecular and cellular levels not apparent in stone fossils. It seems, however, that evolution marches ahead in some lineages while leaving others largely unchanged for millions or even billions of years.

Several lines of argument have convinced some evolutionary biologists that the appearance of progress is an illusion. First, as mentioned before, many lineages hardly seem to change over enormous periods of time. Cyanobacteria (photosynthetic bacteria like cells), horseshoe crabs, and cockroaches, to name a few, seem pretty much unchanged from organisms recorded in fossils that are hundreds of millions of years old, or in the case of cyanobacteria, billions of years old. Any theory that proposes a built-in drive for increasing complexity must account for this. Also, there are alternative explanations. The easiest to understand was famously promoted by Stephen J. Gould, as the "left wall effect."<sup>7</sup>

The left wall effect is a straightforward statistical argument based on the concept of random change. Suppose, for the sake of argument, we could characterize organism complexity by a single number and plot this number for many organisms on a single graph showing lower complexities on the left and higher complexities on the right (figure 12.2).



*Figure 12.2.* The left wall effect. Organisms having low complexity are plotted to the left and organisms with high complexity to the right. Panel A: Dots represent three organisms plotted according to their complexity. A' and A'' are offspring of parent A. Panel B shows parent and offspring early in the history of life. Organism A'' is left of the left wall and therefore dead. Panel C represents a later time and shows a steady spread to the right that occurs over time.

No matter how one defines complexity there must be some lower limit below which no organism can survive. Call this lower limit the “left wall.” When a random mutation occurs in a living lineage, the resulting offspring must have higher or lower complexity than (or be the same as) its parents. Figure 12.2, panel A, shows a parent organism having complexity A with two mutant offspring, A' and A''. Of necessity, they must either fall to the left or to the right (or be located at the same position) on the graph as the parent. Over time, a growing cluster of descendants will form with members both to the left and right of the original parental position. The average of all descendants will stay the same, but the extremes will spread further and further both right and left.



Now consider life's beginning (figure 12.2, panel B). Shortly after the first life arose, organisms must have been very simple and if plotted on the graph would have plotted very close to the left wall. As their offspring appeared to the right and left, those to the right would have thrived while those to the left of the wall would have died (for lack of sufficient complexity). In panel B such an offspring is A". If one were to make a movie showing the location of many organisms on this graph over many generations, the dots (representing organisms) would appear randomly around the first parent. But, because the left-most offspring (left of the left wall) are always removed, the long-term effect would be for the dot pattern to gradually spread out to the right (see panel C). If one measured only the right-most organisms, one would note a steady increase in complexity over time. This is exactly what the fossil record seems to show!

The left wall effect is called a diffusion model. Along with other similar ideas these are what Daniel McShea categorizes as undriven, or passive, mechanisms for generating increased complexity.<sup>8</sup> When a passive mechanism is at work, the observed increases result directly from random change. There is nothing in such models that says anything about what particular features or which individual lineages will increase in complexity over time. They simply predict that there will always be some organism that is most complex and that the measured complexity of whatever happens to be most complex at a given time will increase over time.

A different line of criticism centers on the fact that it has proven very difficult to find numerical measures of organism complexity that correlate well with subjective notions of evolutionary advancement. Given some knowledge of molecular biology, it would seem that the amount of DNA in its cells should provide a straightforward measure of an organism's complexity. About 1970, when technology advances allowed the accurate estimation of this number, it quickly became apparent that there is no simple relationship between the amount of DNA in an organism's cells and its apparent "evolutionary advancement." This became known as the C-value paradox. Figure 12.3 shows ranges of haploid (one copy) DNA contents for a variety of groups.<sup>9</sup>

Although some bacteria (eubacteria in figure 12.3) have the least DNA, there is no obvious reason why certain protozoans should have the most DNA and why some plants and amphibians should have thirty

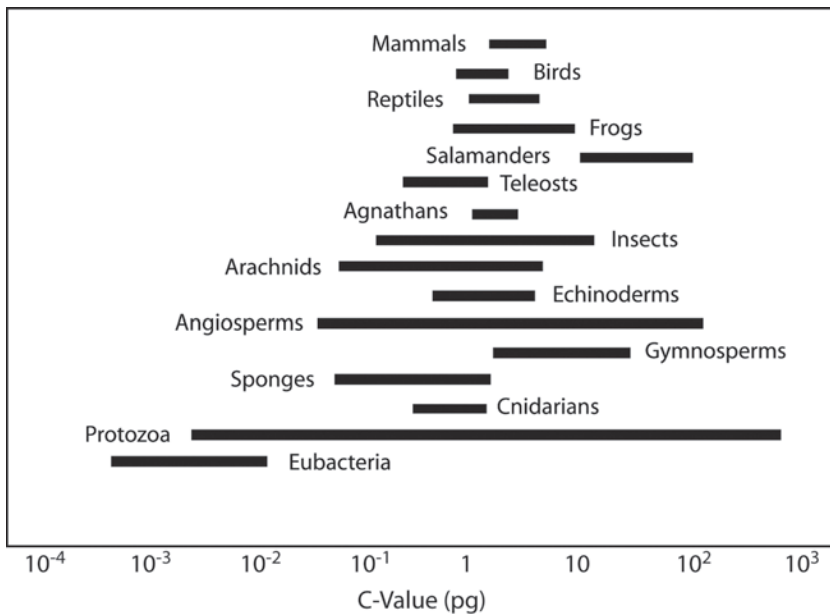
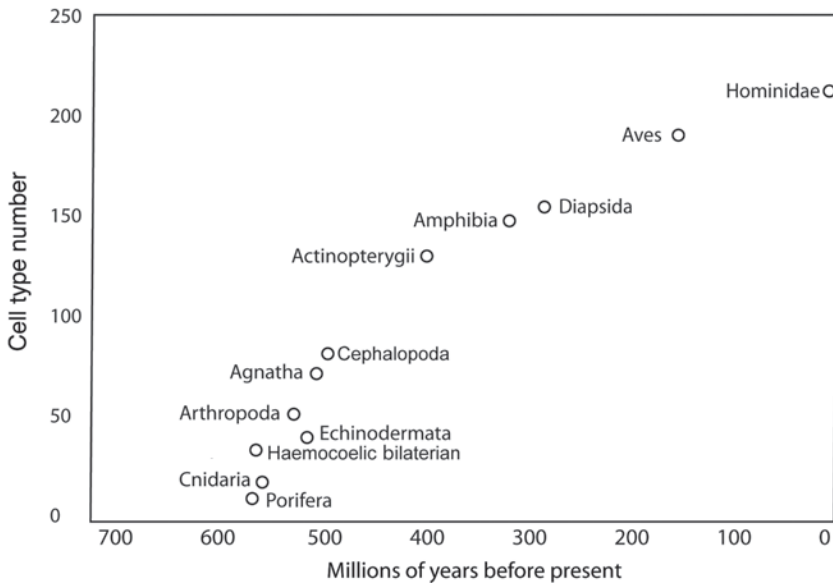


Figure 12.3. A plot of the amount of DNA (the C-value) in a haploid cell for a large number of organisms against some rough notion of evolutionary “advancement.” Long bars indicate that species in this category vary widely. The C-value paradox refers to the obvious lack of correlation between “advancement” and the amount of DNA. Based on a figure presented in T. R Gregory, *Paleobiology* 30 (2004): 179–202.

times more than mammals. Humans have roughly the same amount of DNA as other mammals, while birds have less and salamanders more. The last thirty-five years have seen steady advances in our ability to characterize DNA, first revealing that much of the DNA in cells does not code for proteins, and more recently, direct determination of the total number of genes present in an organism’s DNA. One of the most surprising findings of the human genome project was revelation that humans have about the same number of genes as nematodes and wall cress, a simple flowering plant in the mustard family. It is true that fungi and bacteria have fewer genes than we humans do, but a lot of other species have more.

In contrast to this puzzling finding, a striking correlation has been found between the apparent time of appearance of major animal



*Figure 12.4.* Estimated number of cell types of primitive members of selected metazoan taxa plotted against the estimated time of origin of the body plan of each taxon. Reproduced with permission from J. Valentine. Key: Porifera are sponges; Cnidaria are corals, hydra, and jellyfish; Echinodermata includes starfish and sea urchins; Arthropoda are insects and crustaceans; Amphibia are frogs and salamanders; Aves are birds; and Hominidae are human beings. (see note 10)

groups in the fossil record and the number of cell types present in modern representatives of those same groups. This relationship is shown in figure 12.4.

These data have been criticized on the grounds that categorization of cell type is subjective and that human histology is much more intensively studied than invertebrate histology. Yet, the graph can't possibly be so biased that the trend shown in figure 12.4 is a complete artifact. How could there be a steady increase in complexity as measured by numbers of cell types yet no increase in complexity as measured by gene number? No one knows for sure, but there is at least one intriguing idea for reconciling the observations.

Chapter 9 described a cell as a network of interacting genes and proteins. It is easy to show that networks with the same number of nodes can differ greatly in the complexity of the interactions that connect

their nodes. Perhaps gene number simply tells us the number of nodes while the number of cell types serves as a crude measure of the complexity of the underlying network of interactions. The idea is that every organism is made of cells that are states of the same underlying network. More complex networks would permit larger numbers of stable states, and each cell type represents a particular stable state of the network. By implication, the more stable states that are observed (indicated by the number of cell types), the more complex the underlying network and by implication, the organism.<sup>11</sup>

A recent finding is that a much larger fraction of human DNA is devoted to gene regulation than is the case for yeast or bacteria. Thus far, no one has devised widely accepted methods for identifying and measuring the information content of regulatory DNA. So, we can't yet make graphs plotting this aspect of DNA usage. In any case, the tripartite biochemical, genetic, protein interaction network described in chapter 9 includes more than gene regulation. It is overall network complexity that we should be most interested in. Most likely, a quantitative measure of an organism's network complexity will not be devised until its network can be accurately modeled in a computer. The complexity of the model could then be characterized. We are still some years away from this eventuality.

Daniel McShea has documented a small number of examples showing clear increases in physical complexity that historically have occurred in specific lineages. Such examples are surprisingly difficult to find. The better documented ones include the number of leg types observed in free-living aquatic arthropods over a span of 500 million years, an increase in the maximum number of coiling parameters needed to describe changes in gastropod (snail) shell geometry over hundreds of millions of years, and a statistical increase over 300 million years in the fractal dimension of sutures connecting chambers in the shells of extinct mollusks known as ammonites. A clear example of decreasing complexity is provided by measuring the variability of vertebrae accompanying the evolution of whales from their terrestrial ancestors. In this case, it seems that aquatic life makes simpler demands on spinal column architecture, and evolution apparently favored more uniform vertebrae in whales than in their terrestrial ancestors.<sup>12</sup> Many examples of complexity decreases have been documented in lineages that become parasitic. The evolution of a parasitic lifestyle often is accompanied by a

decrease in apparent structural complexity. This is presumably because the environment of a parasite is less complicated than the environment of a free-living animal.

A different way to approach the problem of complicatedness is to start with the engine of complexity and ask what we can learn about the generation of complexity by examining the underlying computation.

### How do optimization and coevolution fit in?

When considering the broad spectrum of activities that are known to be based on the engine of complexity, we encounter two distinct situations. In one, the environment and selection criteria are unchanging over time. Here, evolving entities become better and better matched to their environments and selection criteria. Sometimes the best possible solution or match is found; other times it is only approached, but always there does exist, at least in principle, a best answer. In the second situation, the environment continually changes; so “best” solutions are transitory. The most interesting systems in this category are those in which the environment of each entity is created, all or in part by other evolving entities. As each entity evolves, the rules of selection change for all other entities. It is sometimes convenient to refer to the first situation (progress toward a fixed best solution) as *optimization* and to use the term *coevolution* for situations in which competition exists among evolving entities. Both situations fall under the broad engine-of-complexity umbrella. Evolutionary systems only reach their full potential for generating novelty when they are coevolutionary.

The simple one-max computation discussed in chapter 6 that converts any sequence of 0s and 1s into a sequence of all 1s is an optimization. The final answer is always the same even though the path to that answer may be different each time the computer algorithm is run. For more complicated tasks the optimal answer may not be known; but in principle, there exists a best answer, and a suitable evolutionary algorithm should eventually find it. After running for a while, if an optimization is stopped before finding the best answer, it is likely to be working with pretty good solutions, and if allowed to run longer will find even better ones. The immune system responding to an infection is an optimization in which the best solution is not known, but the

outcomes generated after ten days are usually good enough to protect the body from further infection. It takes this long for the body to evolve antibodies that bind tightly to specific foreign antigens. One reason HIV, the cause of AIDS, is so difficult for the immune system to clear from the body is that the virus evolves at a rate as fast as the immune system responds; so antibodies produced by the body continually miss their mark. Algorithms that employ the engine of complexity can usually, but not always, solve complex optimization problems. The rate of progress toward an optimal solution is determined by the nature of the problem and by details of the algorithm used. Generally, problems that cannot be solved by an evolutionary approach are characterized by adaptive landscapes in which best answers are not surrounded by other pretty good answers in solution space. Such situations are easily created by computer scientists, but appear rarely in nature.

A general property of optimizations is that no global conclusions can be drawn concerning whether outcomes tend to become more or less complex. The complexity of the outcome depends on whether the optimal solution is complex or simple. Optimizations will create complex solutions when these are needed to do the job and create simple solutions when they work best. Every problem comes with its own expectation.

Returning to the evolution of life, it is usually the case that selection in nature is based on interactions with other organisms. In other words selection is coevolutionary. In such situations details governing selection continually change as organisms evolve and there can be no permanent optimal solution. Evolutionary algorithms can be designed so that *in silico* (computer code) entities experience challenges created by other entities, much like in biology. Any mutation or random change that alters even one interaction changes the overall dynamic. A change in one entity changes the selection dynamic for other entities. In such a situation an endless challenge-response-challenge cycle is established. The adaptive landscape experienced by each entity changes over time as all the entities evolve. In this slowly changing but ultimately unpredictable world, changes advantageous at one time may be disadvantageous at another time, and no innovation assures long-term success. One might imagine that increases in complexity in such systems would not be favored because the myriads of interaction details associated with complexity would be hard to change in a coherent way, and no

particular aspect of complexity would be favored over the long term. But it turns out this is not the case.

One of the great opportunities afforded by evolutionary computer algorithms is that they allow experiments to be performed that are impossible with living organisms. In one type of computer experiment, entities compete for many generations, and sample populations are periodically saved, for example, every 1,000th generation. Populations that have evolved for different lengths of time are then placed in competition with one another. In such an experiment, generation 1,000 can be placed in competition with generation 10,000. In biology this would be the equivalent of collecting trilobites and introducing them into the modern tropical ocean or, alternatively, introducing modern fish in a Cambrian period sea.

The general finding of this type of computer experiment is that populations that have evolved for the longest time generally outcompete those with shorter evolutionary histories. This is true even if populations are taken from different computer runs so that the two populations have no common history. It is as if prolonged competition favors the adoption of general strategies for dealing with new situations. Dan Ashlock at the University of Guelph calls this phenomenon *nonlocalized adaptation*. He has documented six examples based on very different algorithms and very different data structures, and except for some trivial cases where the generation of variation is severely restricted,<sup>13</sup> he has yet to find a system that does not exhibit the effect.

Here is one example: Two of Dan's students, Elizabeth Blankenship and Jonathan Gandrud, wrote a computer program in which two virtual robots (paint-bots) painted squares on a virtual  $12 \times 12$  board. Each paint-bot used a different color of paint and painted a square by occupying it. The bots took turns moving and painting one square at a time. Each bot could detect the colors of the eight squares immediately adjacent to it. Paint-bots were allowed only three possible moves, turn right, turn left, or go forward. The board was divided into 144 squares to be painted, and each game consisted of 288 moves by each bot, enough for each to paint the entire board twice. The winner of a game was decided by counting the number of squares of each color after 288 moves. In this game, a paint-bot that systematically paints all the squares on the board may lose the game because the other bot may simply follow it painting over each opponent-painted square. Paint-bot

movements were determined by a piece of software called a controller. The controller accepts as input the colors of surrounding squares (these being unpainted, the opponent's color, or its own color) and decides a next move. In the experiment, populations of bots (actually controllers) were evolved for a large number of generations where each generation consisted of a prescribed number of contests between pairs of paint-bots.

In the specific experiment reported there were 200 controllers (paint-bots), and in every generation each paint-bot in the population played four randomly selected opponents five times. After every generation, the scores were tallied, the 100 controllers with the best total scores were duplicated, and their offspring mutated and paired for recombination. Recombination consisted of randomly breaking the controllers (strings of computer code) at the same distance from one end and joining the left half of one with the right half of the other to create two new controllers. The recombined and mutated controllers were then introduced into the player pool along with the 100 winning parents and another tournament (generation) played. In every cycle, the 100 lowest-scoring robots were replaced by mutated offspring of the 100 best, and the 100 best survived unmodified for another round.

Repeated playing of this computer game revealed that when initial controllers specified random moves, the average painting scores (number of squares painted by a bot) increased rapidly for the first twenty to thirty generations and then leveled off. After about fifty generations, the number of squares painted was typically about sixty (out of 144) and did not increase consistently after that, even if thousands of generations were played. In the initial phase, totally incompetent painters were eliminated and the survivors exhibited distinct strategies. Remember that 288 moves is enough for a simple repetitive painting strategy to paint all 144 squares twice; so an average score of 60 suggests that successful bots spent most of their time painting over one another's paint jobs! The biological analogy of this would be a creature that spent more time foiling its competitors than looking for food.

The most interesting aspect of the experiment was revealed when bots from generation 500 were pitted against bots from generation 5,000. When this was done, the more highly evolved paint-bots outscored the less evolved bots over 99 percent of the time. This is in spite of the fact that paint-bots from both generations score about 60 when playing



bots from their own generations. Statistically, the odds that the observed results occurred by chance are less than one in 10 million ( $10^{-7}$ ).

This and similar results in other very different systems suggest that entities in more evolved populations are more complex in some way than those in less evolved populations. In the paint-bot example, the instructions (the controllers) that determine entity behavior did not grow appreciably in length; so complexity measured by the length of the instructions did not increase. What seems to be happening over time is that strategies successful in a broader range of situations are found. These broader-ranging strategies seem to be effective even in situations the population has never before encountered. The longer-evolved instructions are somehow more versatile than those employed in earlier generations. Scores do not improve over time because the competition gets tougher. One could say that all members of the population become more and more clever. It is hard to see why a similar phenomenon would not be at work in biological populations.

### Do resources matter?

As a general rule, complexity has associated costs. The notion of complexity, unless the term is defined as randomness, implies the presence of structure; and creation of structure does not come free. A powerful but general way to analyze the costs and benefits of complexity is to associate success of entities in complicated situations with their ability to maintain control. The logic of this is easy to see in biology where death results when an individual's internal environment is not maintained within certain limits.

The study of control as a formal branch of engineering dates back, at least, to the invention of the steam engine. In 1956 W. Ross Ashby put forward what he called the *Law of Requisite Variety*. This states that "In order to achieve control, the variety of actions a control system is able to execute must be at least as great as the variety of environmental perturbations that need to be compensated."<sup>14</sup> This principle is used widely by engineers designing electronic and mechanical control systems such as might be found in heating and air conditioning systems or to keep a rocket on course. Ashby was very cognizant of the biological implications.

Phrased in the terminology of complexity, Ashby's law of requisite variety says that in order to assure internal stability, a controller must be at least as complex as the environment it contends with. When applied to biology, this principle implies that evolution should tend to favor those organisms that are best able to maintain control over their internal and external environments. Not every aspect of the environment on Earth is relevant to every organism. The environmental complexity important to any particular organism depends on its lifestyle and scale. What is important to a soil bacterium is quite different from what is important to a tiger. Because environmental complexity is so multifaceted and organism specific, no universal method for measuring it is widely accepted in the scientific community.

A logical consequence of Ashby's principle is that it is quite impossible for any one organism (or controller of any kind) to achieve total control in a system that consists of diverse competing entities when each has a similar level of complexity. This is because the complexity of all the other competing entities must be greater than that of any one. The resulting control shortfall ought to provide continual selective pressure favoring ever greater single entity complexity. The inherent conflict between advantage due to increased complexity and costs associated with increased complexity must provide special advantage to cost-efficient features and for features that are versatile in the face of multifarious and unexpected environmental challenges.

For systems characterized by unpredictable environmental change, other things being equal, favored entities will tend to evolve structures and strategies that succeed under a large number of conditions. In the face of unpredictable environments, entities with more capabilities have a higher probability of success than entities with fewer capabilities.

Working against any unrestricted growth of capabilities is the reality that every capability has associated costs. As the number of capabilities increases, so do the costs. Entities that have been evolving for a long time must reflect a balance between costs and benefits. In the physical world, four broad cost categories always come into play. These are energy, entropy, time, and space.

Of these, energy is the most familiar. Your automobile will not run without gasoline. Our bodies will not run long without food. In general, any physical device that does something uses energy. In biology, some energy costs are obvious: organisms move, transport molecules

across their membranes, create new structures, and repair old ones. The requirement for energy is the reason animals must eat and plants must have sunlight. For genetic algorithms, energy costs are no less real. Computers must be plugged in. Recording data to a hard drive or retrieving it from computer memory takes energy. Another ever-present demand is maintenance. In general, the greater the number of parts and the more organized those parts, the higher is the cost of maintaining the structure. Energy used for maintenance holds entropy at bay.

Entropy measures disorder, and organized systems have a natural tendency to become more disordered. A central principle of thermodynamics is that energy can be used in ways that oppose this natural tendency toward increasing entropy. A good example is maintaining a neat desk. As we work we must always expend energy to keep our desks neat; otherwise the papers, books, pens, paper clips, and Post-it notes pile up. In the absence of maintenance effort, any organized system will eventually decay into disorder. Fundamentally, the engine-of-complexity cycle runs counter to entropy because it computationally extracts purposeful order from disorder. The order gained is in the form of purposeful structure. As evolutionary systems accumulate more and more structure, they become more and more susceptible to damage by random change and require more and more energy for maintenance. Any system that becomes more and more complex faces a reality that the energy cost of maintenance must eventually equal any benefit provided by further increases in complexity.

Time costs are less obvious, but surprisingly important. If relationships between an entity and its environment are viewed in terms of challenges and responses, then many situations emerge where advantage goes to the quicker response. When a lion encounters a gazelle, speed is all-important and not just running speed, but also decision-making speed. A more subtle aspect of time cost comes about because some challenges are better met by complex responses than by simple ones, and complex responses take time to compute. Other things being equal, there is a trade-off between response complexity and response speed. The gazelle may be able to evade the lion by being clever, but if it takes too long to ponder its next move, the outcome is not likely to be good for the gazelle.

An underappreciated aspect of the time required to compute complex actions and also to compute complex structures is that the required

time can sometimes be spent in advance. Contrast the encounter of a lion and a gazelle with an encounter between a lion and a prepared human. The human cannot outrun the lion, but may carry a weapon. The gun or spear took much time to create, but that time was spent before, not after, a lion is encountered. The complexity inherent in the weapon is created and banked for future need in the form of *potentially useful structure*. Situations like this occur often, and it is common to find structure that required a time-consuming process to produce but is then available for some future fast response. The aspect of complexity that permits fast but computationally intensive responses is called *depth*.

Space is also an essential resource, but in the physical universe it is not usually limiting. In computer science space in the form of limited memory has until recent times been a precious resource, but as mentioned before, limited space often can be compensated for by expending more time.

### How does depth figure in?

Depth provides one formal measure of the intricacy of the internal relationships that exist within a computational object.<sup>15</sup> The basic idea is this: the more complicated the internal relationships, the more steps it takes to compute an output having those complications from input that doesn't have them (i.e. is simple). This property is known as "slow growth," and it means that deep objects can be created quickly only from deep inputs. Creation of complicated (deep) objects from simple (shallow) inputs requires a lengthy computation. There are no shortcuts to this rule. Since physical processes can be formally treated as computations, physical objects also must have a depth-like property.

Extending this concept to interactions between agents is straightforward. When challenges and responses are viewed as computations, a deep challenge is one that requires extensive computation to create, and a deep response is one that requires extensive computation to produce.

The slow-growth property provides a powerful rationale for why deep entities might be favored in some evolutionary situations. In nature organisms face innumerable time-dependent challenges. Because preparations can be made in advance and the results stored in the form of

deep structure, it is possible to create or compute, in advance, structures that allow quick but computationally intensive responses to be made later. Carrying a gun in the wilderness is an example. Shallow responses by definition must be random or highly repetitive, and random or repetitive actions are often not the most effective. Because of this, deep structures that yield rapid complex responses should be favored by selection whenever quick complex responses are advantageous and the associated costs are not too high. It is plausible that animal brains and elaborate metabolic networks are examples of deep biological structures created over time by long-term selection favoring rapid effective responses to deep challenges. They are also structures that provide increased control.

Extensive computation is a waste of time and energy if a challenge is effectively met by a simple response. Predictable challenges typically require only memorized responses (simple computations). Totally unpredictable challenges cannot be planned for at all. In this case the best option is to respond randomly (also computationally simple). In between are a huge variety of challenges that can be best met by computationally intensive action. Biology is filled with examples. Think of a cat hunting for mice. The cat needs food or it will die. Mice live in holes and forage mostly at night when they are hard to see. In response to this challenge, cats have evolved quick-acting muscles, eyes that see well in low light, excellent hearing, and a brain that allows them to predict where mice are likely to be encountered.

While some challenges demand only simple responses, it must be that many are better met by responses requiring extensive computation. In particular, there is a tendency for deep challenges to be met optimally by deep responses. But how do deep challenges arise? In the biological world the physical environment experienced by an organism is probably not very deep. Most aspects of the physical environment are either random or repetitive. Examples are the day-night cycle and the fact that it will be cold in winter and warm in summer; day-to-day weather details are effectively random within a broad climate context. If organisms needed only to contend with the physical environment, we should not expect them to evolve particularly deep structures.

In contrast, when the environment contains evolving agents that compete with one another, the situation is different. Each entity experiences challenges created by other entities. Every action creates a

challenge for another. Consider cats versus mice. Over their long history of competition, mice have gotten better at avoiding cats, and cats have gotten better at finding mice. As evolving agents challenge and respond, selection always will favor more effective responses and penalize those that fail to keep up. But, as each measure is met by countermeasure, the overall “success” of one population when compared to others may not change at all. This is the “Red Queen” principle first articulated by Leigh Van Valen at the University of Chicago.<sup>16</sup> Since an important aspect of effectiveness is timeliness, whenever more effective responses are both computationally intensive and time sensitive, challenges and responses will tend, over time, to become deeper and deeper. This positive feedback loop would seem to guarantee the evolutionary creation of deep structure. If we accept depth to be an aspect of complexity, then coevolution should lead to its increase.

The same feedback loop should be present in nonliving evolutionary systems whenever there is competition among evolving entities and response time is important. This leads to the clear prediction that complexity, as measured by depth, should increase in various evolutionary systems featuring competition among evolving entities, unless response time is unimportant or when the energy cost associated with the resultant complexity is too high.

### Is there evidence?

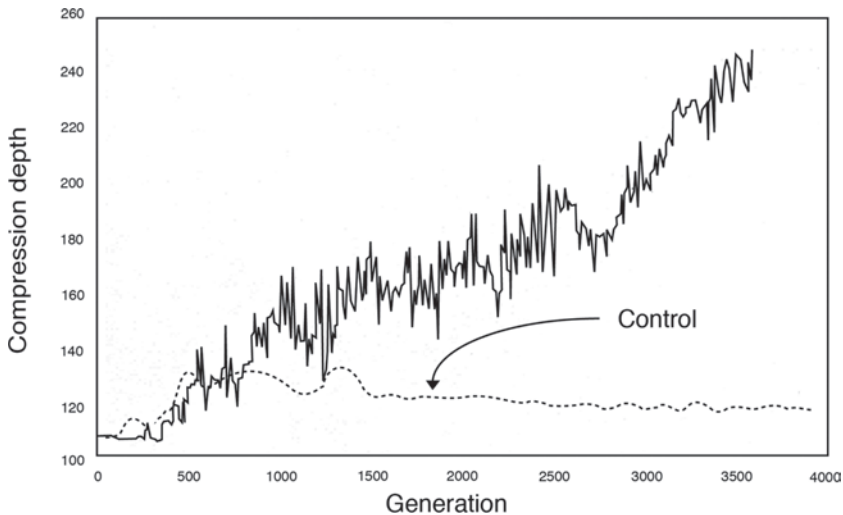
In science, for a prediction to have value it must be testable. This is no less true for the hypothesis that depth is an important aspect of complexity generated by coevolutionary systems. A big problem with the computational concept of depth is that it is not computable—it is a measure that can’t be measured! Algorithmic information, introduced in chapter 2, is another computational measure with this same disquieting property. Fortunately, both can be approximated.

Conceptually, depth characterizes how difficult it is to compute something from simple input. The reverse computational procedure, compression, creates a short file from a larger input without losing information. Compression is commonly performed in computer applications where large files are a problem, such as storage on a hard drive or transmission over the Internet. When a file is compressed reversibly,

the original file can be recreated by a decompression computation. JPEG and ZIP files are compressed. Compressed files must always be expanded before they can be used. “Compression depth” is a computable measure of the difficulty of creating a compressed version of a digital object. It is believed by theoreticians that most, but not all, difficult-to-create (i.e. deep) objects are also difficult to compress.

In 1997 James Lathrop, then a graduate student in computer science at Iowa State University, performed several computer experiments in which binary entities were evolved in an environment of other evolving entities. The task he explored involved “finite-state machines,” which are a category of computer entities one can classify as agents because they make decisions based on their current situations and remembered past actions. He had these agents play a game called iterated prisoner’s dilemma, and he evolved them by means of a genetic algorithm that selected agents according to their game scores. As expected the finite-state machine agents evolved better and better strategies for playing the game. Because the agents were actually computer files, Lathrop could measure how difficult it was to compress them. The specific measurement he used was based on the widely used compression strategy known as L-Z compression. Figure 12.5 shows the increase in compression depth exhibited by agents during this exercise. As you can see, the compression depth of the agents steadily increased over the course of 3,500 generations with no apparent slowdown over time.

A straightforward interpretation of this experiment is that L-Z compression depth measures various aspects of the organizational complexity of the agents. The more layers of entangled organization that are present, the more demanding is the computational task of uncovering and re-encoding them into a more compact form. Compression depth measures the amount of computational resources (memory in this case) required to accomplish a given amount of compression. An important feature of this experiment was that the agents (finite-state machines) did not get larger (require more computer code) over time, rather they seemed to become more cleverly organized. The experiment is repeatable and the phenomenon has been observed for different data structures and different selection tasks.<sup>17</sup> The changes observed may be related to nonlocalized adaptation discussed earlier in this chapter. Regardless, both observations make clear that changes occur in the internal structure of computer entities evolving in coevolutionary



*Figure 12.5.* Accumulation of L-Z compression depth during an adaptive process. Reproduced with permission from J. Lathrop (see note 17).

situations that are not captured by the concepts of Shannon or algorithmic information. It is likely that similar phenomena can be observed in other coevolutionary systems.

A wonderful experiment would be to compare the compression depth of DNA sequences from organisms that lived long ago with DNA from modern organisms. Unfortunately, this is impossible because fossilized DNA does not survive for the tens or hundreds of millions of years necessary to perform the experiment. The shorter times that can be examined (e.g., DNA from ice age man or from mammoths) are too short to permit detection of a difference. The ultimate experiment, of course, would be to compare the depth of DNA from organisms that existed very early in the history of life with DNA from those that came much later. Since this isn't going to happen, we must be content with characterizing the output of evolutionary computer algorithms and inferring what must also be happening in the biological realm.

Depth is a computer science measure not yet widely seized upon by noncomputer scientists who write about evolution, partly no doubt because it cannot be measured. Lathrop's and Ashlock's experiments show unequivocally the existence of aspects of complexity that increase



in some coevolutionary systems that are not captured by more traditional measures of organization. Theoreticians who study depth have variously characterized the property as: “value inherent in an object,” “computational usefulness,” “the amount of computational history evident in an object,” “organization of redundancy,” “intricateness,” “computational work stored in organization,” and “organization embedded in an object by computation.” The common thread of these verbal descriptions of an abstract mathematical concept is that information can be organized by the process of computation to create output structures that exhibit increased usefulness when they serve as inputs to future computations.<sup>18</sup> An important way that usefulness is manifest is speed. Many contexts exist in which fast complex responses are useful. Structures that make this possible are deep.

The existence of this property raises the intriguing possibility that when people perceive complexity, part of what they perceive is depth. One property of deep objects is that they must have a long history. A gnarled old tree, a human brain, and a mature ecosystem provide examples of things that have a required long history. None could possibly have been created quickly. This is the hallmark of depth. Perhaps perceived depth is why we tend to appreciate and value things that have a long history.

---

## Past and Present

### How do engines of complexity get started?

The two most frequently asked questions I get from students who are first learning about evolution are: How did life start? And, what is its future? These are hard questions. The engine-of-complexity concept provides added perspective to both. I will start with origins.

If you think back, we have considered four examples of the engine of complexity operating today on Earth: life, the mammalian adaptive immune system, social change (with religion, science and technology, and the economy as subexamples), and evolutionary computer algorithms; with human learning presenting a possible fifth example. Every implementation of the engine requires a physical system to encode information and carry out the computation. Our next task is to compare the origin of each with the goal of finding common features.

### THE ORIGIN OF LIFE

Much has been written about the origin of life, but a satisfying scientific understanding of this event has been difficult to achieve, mostly

because direct evidence is so scanty. Some things are known with high confidence. First, there are only three general ways, even in principle, that life could have arisen on Earth. Life could have: (1) come here from somewhere else in the universe; (2) arisen on Earth through the action of some power acting outside the known rules of chemistry and physics; and (3) arisen on Earth through natural chemical and physical processes. The third possibility can be further subdivided into scenarios having a reasonable probability and those that are ridiculously improbable (i.e., a miracle). There is currently no evidence supporting the idea that life originated elsewhere, but the discovery of DNA-based life on Mars or Europa (a moon of Jupiter) could dramatically change that, and origins inconsistent with the known laws of chemistry and physics are not subject to investigation. So, scientists are left with no choice but to investigate chemical scenarios that might have occurred here on Earth, although NASA continues to fund experiments designed to detect life on Mars and elsewhere in the solar system.

Fossil and geochemical evidence indicates that life arose on Earth sometime before 3.5 billion years ago (bya) but no earlier than 4.2 bya, with weak evidence hinting 3.8 bya or before. The oldest surviving rocks formed about 3.8 bya, though there are crystals called zircons that are older. Considerable uncertainty surrounds conditions that prevailed on Earth in this era. Standard solar models indicate that the sun was only 75 percent as bright as today, implying that the oceans should have frozen over, but the presence of greenhouse gasses, either carbon dioxide or methane, and a much warmer planetary interior, could have warmed the surface enough to keep the oceans liquid. There must have been a lot of volcanism releasing gasses. In fact, most early Earth models have much or all of the atmosphere originating from such releases. Hydrogen, carbon, oxygen, nitrogen, and sulfur were all present, but controversy surrounds what chemical forms these elements took.

The single largest unknown is the oxidation state of the atmosphere. The modern atmosphere is strongly oxidizing (lots of oxygen), but scientists agree this is the result of photosynthesis. Life maintains molecular oxygen ( $O_2$ ) in the atmosphere by splitting water ( $H_2O$ ) and releasing oxygen. The energy required to do this comes from sunlight. Molecular oxygen is chemically reactive, and in the absence of photosynthesis would rapidly disappear from the atmosphere. Under the current oxidizing conditions, carbon occurs as carbon dioxide ( $CO_2$ ),

hydrogen as water ( $\text{H}_2\text{O}$ ), nitrogen as a gas ( $\text{N}_2$ ) or as nitrate ( $\text{NO}_3^-$ ), and sulfur as sulfate ( $\text{SO}_4^-$ ). Sulfate and nitrate are found dissolved in water or bound up in rocks. The opposite end of the oxidation/reduction chemical spectrum is strongly reducing. This is where atoms are bound to hydrogen rather than oxygen. Under such conditions, carbon occurs as methane ( $\text{CH}_4$ ), nitrogen as ammonia ( $\text{NH}_3$ ), sulfur as hydrogen sulfide ( $\text{H}_2\text{S}$ ), hydrogen is in its molecular form ( $\text{H}_2$ ), and oxygen is in water ( $\text{H}_2\text{O}$ ). In-between conditions are characterized by water, molecular nitrogen, carbon dioxide or carbon monoxide ( $\text{CO}$ ), and elemental sulfur (a solid). A variety of other compounds including formaldehyde ( $\text{CH}_2\text{O}$ ) and cyanide ( $\text{HCN}$ ) are likely possibilities.

Geochemical views about the composition of the early atmosphere have changed over the years as more has been learned about geologic processes and the makeup of the oldest rocks analyzed in greater chemical detail. Current models favor neutral or moderately reducing conditions ( $\text{H}_2\text{O}$ ,  $\text{N}_2$ ,  $\text{CO}_2$ , possibly  $\text{CO}$ ,  $\text{S}$  and/or  $\text{H}_2\text{S}$ ) with little molecular oxygen. Near volcanic vents, conditions would have been locally strongly reducing, just as they are today. With light from the sun, wet and dry cycles, frequent lightning, and lots of volcanic activity, a significant amount of chemistry involving hydrogen, carbon, nitrogen, oxygen, and sulfur would have occurred leading to the formation of a host of carbon-based molecules. A similar process may be exhibited today on Titan, the large moon of Saturn that has a strongly reducing atmosphere containing methane and ammonia. Its atmosphere is a hazy orange color due to the presence of complex organic molecules based on carbon, hydrogen, and nitrogen. Titan is very cold, so oxygen is mostly tied up in frozen water.

The most commonly quoted candidate theory for the origin of life dates back to Darwin, who in 1871 wrote to Joseph Hooker:

It is often said that all the conditions for the first production of a living organism are present, which could ever have been present. But if (and oh! what a big if!) we could conceive in some warm little pond, with all sorts of ammonia and phosphoric salts, light, heat, electricity, etc., present, that a protein compound was chemically formed ready to undergo still more complex changes, at the present day such matter would be instantly devoured or absorbed, which would not have been the case before living creatures were formed.

Today, Darwin's "warm little pond" is more likely to be referred to as the "primordial soup," a bouillabaisse of amino acids, sugars, nucleotides, and a lot of other organic chemicals dissolved in water happily reacting with one another in wild abandon. This cheerful vision has a variety of problems that have not been solved. Foremost of these is that although most key chemicals can be created in the laboratory under plausible conditions, the discovered conditions are not the same for all classes of chemicals. Thus scenarios must be imagined in which different chemicals were created in different ponds and then combined in just the right order. Another problem is that for polymers to occur (proteins, polysaccharides, and nucleic acids are all polymers), the "soup" needs to be rather concentrated, something that could not have happened in the entire ocean because organic chemicals characteristic of life are chemically labile in the presence of excess water and would break down as fast as they were made.

A host of ideas have been offered to solve these problems including concentration in drying ponds, concentration in liquid pockets formed in ice, specialized chemistry on the surface of catalytic clays, specialized environments in contact with hot rocks such as would occur in underwater volcanoes, chemistry in sediment (mud), and chemistry in the solar nebula that was later delivered to Earth by comets and meteorites. Each of these ideas open up new dimensions to be explored, but in truth no one has yet demonstrated conditions in the laboratory that provide all the appropriate chemicals and are also judged plausible given what is currently surmised about conditions that prevailed on the early Earth.<sup>1</sup>

Regardless of the chemical details, everyone who studies the issue in depth agrees that two critical achievements were necessary for life to arise. An interlocking self-sustaining network of chemical reactions needed to arise, call it the precursor to metabolism, and the catalysis and regulation of the various chemical reactions that make up the network needed to come under control of instructions. Catalysis is needed because so many modern metabolic reactions do not readily occur at moderate temperatures. In all forms of modern life, catalysis and control are achieved by the production of protein enzymes whose structures are specified by information encoded in DNA. The engine of complexity operates in modern life because that information is replicated and subjected to cumulative selection.

There has long been a major disagreement among experts concerning which came first, metabolism or instructions. Was it the chicken or the egg? Some insist that the self-replication of an information-encoding molecule must have been the seminal event. The idea is that once self-replication was achieved, expansion of the instructions to include catalysis of other reactions (in addition to the self-replication reaction) would not be very difficult. For many years, the conundrum presented by the observation that DNA codes for proteins and proteins are necessary for the synthesis of DNA was seen as the central problem. Surprisingly, this problem was solved, in principle, in the 1980s without solving the origin of life. The key discovery was that RNA, which is an information-carrying molecule very similar to DNA, can catalyze chemical reactions. Thus, in principle, it is possible for there to be an RNA molecule that catalyzes its own synthesis. No chemist has yet made such a molecule, but if it were to form through random chemical activity, and was immersed in a broth of nucleoside triphosphates, the chemical precursors of RNA, the molecule would spontaneously make copies of itself. Because this hypothesized molecule would encode in its structure the information needed to make more RNA molecules, it would automatically be locked into an engine of complexity cycle, and its descendants would inevitably and quickly become better and better replicators. This concept led in the late 1980s to a vision of an early stage in the origin of life called the “RNA world”—a pool of self-replicating RNA molecules all competing with one another for space and nucleoside triphosphates. Under selective pressure, these RNA molecules gradually “learned” (through the evolutionary process) to encode the production of protein enzymes necessary to catalyze the formation of nucleoside triphosphates from simpler precursors and eventually all the other biochemicals that characterize biochemical networks in modern cells. In this scenario, the structure manifest in intricate biochemical networks is a natural outcome of the engine of complexity operating in a chemical system.

This is a wonderful vision, but in spite of recent breakthroughs no one has been able to provide a believable scenario in which all the needed nucleoside triphosphates are produced in sufficiently high concentrations to polymerize into RNA chains, nor has a self-replicating small RNA been demonstrated.<sup>2</sup> On the other hand, the general concept of self-replicating molecules constituting the essential first step

remains very appealing, and some scientists are looking hard to find conditions under which this might have happened.

The alternative vision, metabolism first, has a longer history of champions. A recent version is offered by Stuart Kauffman.<sup>3</sup> The idea is that the critical first step was not a self-catalytic replicating molecule but rather a self-catalytic self-sustaining network of chemical reactions. He calls it an "autocatalytic set." It seems ingeniously plausible. In a system (soup) of more or less random chemical reactions, some simple polymers will (randomly) form. Monomers will form dimers (two monomers linked together) and trimers (three monomers linked together), and dimers and trimers will form tetramers, pentamers, hexamers, and so forth. The number of possible products expands rapidly as the number of monomers linked together gets larger. Initially, the concentrations of these polymeric species would be very low.

A characteristic of some polymers, including proteins and RNA, is that some of them will catalyze particular chemical reactions. Thus, in a broth of reacting chemicals it is not much of a stretch to imagine that small polymers will be present and that a few of these will individually catalyze some chemical reaction; it doesn't much matter which reactions are catalyzed. Chemical products of the catalyzed reactions will increase greatly in amount (because of the catalysis) resulting in new chemical reactions and new polymers. Whenever a new catalyst is formed, many details of the chemical network change. Any time a catalyzed reaction leads even indirectly to an increase in the concentration of its catalyst, a positive feedback loop is established between the products of a reaction and its catalyst. Of particular interest are small polymers that catalyze the formation of larger polymers, say by accelerating the joining of a particular dimer to a particular trimer. The potential for small polymers to catalyze such reactions has been demonstrated. For example, the very simple polymer diglycine (glycine is the simplest amino acid and diglycine is two of them bonded together) catalyzes the polymerization of several amino acids and small peptides in high salt solutions, such as might occur in drying ponds.<sup>4</sup> The resulting new polymers would be available to join with other small polymers to create still larger polymers, each with some probability of catalyzing some other reaction. If the probability of small polymers catalyzing the formation of larger polymers is high enough, the process leads inevitably to the formation of more and more positive feedback loops.

This whole idea sounds a little fishy, but Kauffman and his colleagues have run computer simulations showing it just might work. By incorporating certain assumptions about how probable it is for a randomly chosen polymer to catalyze a polymer-forming reaction, when the number of chemical species in the initial reaction broth becomes sufficiently large, “catalytic closure” occurs in which the network becomes self-sustaining if supplied with simple chemical feed stocks (precursor molecules).<sup>5</sup> A self-catalytic self-sustaining biochemical network is exactly how I defined life in chapter 9. If Kauffman’s vision is what actually happened, the tripartite biochemical/proteome/regulatory network that defines each cell today is the latest stage of a continuous lineage of biochemical networks that extends back to an autocatalytic set that formed before the first cell and even before the involvement of RNA or DNA. Somewhere along the way, but fairly early on, the network gradually incorporated instructions (quite possibly in the form of RNA) because instructions provide a more effective way of managing and accumulating purposeful information. Instructions are also much easier to evolve.

One appealing feature of this vision is its generality. The required properties are not a function of a particular collection of chemicals, but rather, catalytic closure occurs whenever certain general conditions are met and the number of chemical species present is sufficiently large. This feature is also a weakness. To convince large numbers of scientists, specific chemistry needs to be demonstrated in the laboratory. Exactly what catalytic polymers are plausible given what we know about conditions on the young Earth? Polypeptides (short proteins) are a good bet because amino acids are produced under a wide variety of simulated early Earth conditions, short polypeptides are known to have catalytic activities, and polypeptides are what life uses for catalysts today (cellular enzymes are long polypeptides). Still, no one has yet demonstrated an artificial polypeptide-based autocatalytic set in the laboratory, and until this happens, it is appropriate to be optimistically skeptical.<sup>6</sup>

An interesting feature of the autocatalytic model is that polymers, by their nature, encode information in their sequences about whatever catalysis they might perform. Thus, an autocatalytic set of polymers encodes in the distributed structures of its multiple polymers the information needed for its own propagation. Selection operating alone would have been a powerful force shaping the dynamics and details



that emerged in such a system. If certain network reactions synthesized nucleotides and polymers of nucleotides, then a clear path from the autocatalytic set to the RNA world is not hard to imagine.

The RNA world and the autocatalytic set have been traditionally seen as alternative visions for the origin of life often characterized as “information first” and “biochemistry first.” But, as we have just seen, they need not be in conflict. If the catalytic set came to include RNA, an RNA world could have arisen from an older autocatalytic chemical network that continued to supply it with nucleoside triphosphates.

To summarize, the information-first model requires the random formation of informational molecules, one of which quite by accident catalyzes its own replication. Any self-replicating molecule becomes automatically engaged in the engine-of-complexity cycle (assuming its replication is imperfect). Because specific RNA sequences are known that catalyze specific chemical reactions, it is not hard to imagine how the evolution of such a self-catalytic molecule, if it ever existed, could have led to the construction of a biochemical network that supplied needed precursors required for replication to continue after the initial reservoir of precursors was depleted.

In contrast, the biochemistry-first model envisions that biochemical networks became quite complex before they consolidated distributed catalytic and regulatory information into specialized information-encoding molecules (RNA or DNA). Under this scenario, the incorporation of molecules specializing in the storage and use of information was needed to fully engage the engine of complexity, but was an outgrowth of selection acting on an already-existing biochemical network. With this vision, the first informational molecules were not self-replicating but were synthesized along with all the rest of the chemical components of a self-propagating network.

It is not clear that logic alone can resolve which of these ideas is closest to what actually happened. Such a resolution will only come when one of the two models (or something else) is actually demonstrated in the laboratory under conditions that seem plausible given all that is known about early Earth geochemistry.

An appealing feature of the biochemistry-first scenario is that selection plays an important role prior to any explicit encoding of instructional information, and selection is decidedly low tech. It occurs widely in nature whether or not life is involved. The biochemistry-first model

also seems less dependent on a special low probability event. Once a system with certain properties gets complicated enough, it automatically self-organizes into an autocatalytic set.

With the information-first model, a simple probability calculation places clear and severe limits on the amount of purposeful information a first self-replicating molecule could have encoded. The calculation goes like this: Assume the existence of a system that randomly generates information-carrying molecules at the rate of one per second. This is how long it takes *E. coli* RNA polymerase in a living cell to transcribe one gene. Since it is extremely unlikely a primitive replication system would be so efficient, this is a conservative estimate. Further, assume that a very large number of molecules are synthesized every second. For the sake of argument I will arbitrarily pick this number to be  $6 \times 10^{23}$ . This is known in chemistry as a mole.<sup>7</sup> It represents a very large number of molecules and presupposes a very large source of precursor monomers. Allow this activity to continue for 100 million years ( $2 \times 10^{15}$  seconds). Over this span of time, roughly  $10^{39}$  such molecules will be produced.

If these molecules were RNA (with four chemical residues constituting the information-encoding alphabet), they would collectively encode  $2^{130}$  bits of information. To put it another way, 100 million years would provide enough time for this system to explore an information space of 130 bits. This means that over the span of 100 million years most RNA molecules sixty-five nucleotides long or less would be produced (allotting two bits per nucleotide), but most molecules much larger than this would never appear (assuming random synthesis). If we increased the production to include all the carbon atoms on Earth (a truly absurd assumption), the information space that could be explored would still be less than 200 bits. Because of the exponential relationship between length and number of possible sequences, it would have been completely impossible for a particular meaningful polymer sequence to appear on the early Earth that was more than about 100 monomers long, and fifty is a more plausible upper limit. So, a simple mathematical conclusion is that the first informational molecule, if it was randomly generated, could not have been very long.

One hundred bits (fifty RNA monomers) is less information than is contained in one line on this page or in a short musical jingle. The instruction “boil the egg for three minutes and salt and pepper to taste”

requires about 100 bits of information. The information-first model forces us to surmise that it is possible to construct a fifty-unit polymer that folds into a shape capable of catalyzing its own replication (given a supply of appropriate simple precursor molecules). Such a molecule has not been demonstrated; but this does not mean one cannot exist. In contrast, RNA ligase enzymatic activity has been well studied in RNA molecules. A ligase is an enzyme that joins two existing polymers to create a larger one. RNA ligases could have played a major role in creating the first self-replicating RNA. A major virtue of the biochemistry-first model is that the earliest encoded information need not have been specifically concerned with self-replication, but rather, any kind of information that enhanced the performance of the network would do. In the biochemistry-first model, the probability problem seems to be overcome gradually, requiring no highly improbable steps (except for those initial catalytic closure events whose probability is currently impossible to calculate and for which expert estimates currently range from “near certainty” to “virtual impossibility”!). To conclude this section, the critical event in the origin of life was self-replication of either an information-encoding molecule or an information-encoding autocatalytic network, but it is not possible with current knowledge to pick confidently between them. Both scenarios would have been subject to selection.

## THE ADAPTIVE IMMUNE SYSTEM

The second origin we will explore is the adaptive immune system. All living organisms possess molecular mechanisms for fending off attacks by microbes. Without them, any plant or animal or fungus would quickly become bacterial lunch. A host of defense mechanisms have been identified. Many of these are lumped together under the general heading of “innate immunity.” Included are mechanisms for recognizing and destroying foreign DNA and RNA, mechanisms for recognizing specific polysaccharides and proteins that characterize the surfaces of bacteria, secretion of antimicrobial peptides, and in all animals, amoeba-like cells that patrol the body engulfing and digesting substances that should not be there. The success of plants and animals is testimony to the collective effectiveness of these mechanisms. An inherent problem with all innate defense systems is that microorganisms can

evolve countermeasures faster than larger slower replicating hosts can evolve new defenses. Early in the history of the vertebrate animals, some 450 million years ago, a new strategy was hit upon that when refined by further evolution allowed vertebrates to evolve a new defense within their bodies as fast as or faster than microorganisms can evolve. We call this the adaptive immune system.

Evolution of this system was gradual, involving many intermediate steps. Essential elements of how it works today were described in chapter 7. Creation of new antibody and new T-cell receptor genes specifically tailored to an invader is the activity that most distinguishes adaptive immunity from innate immunity. This system works because two distinct strategies are brought together, rearrangement of existing antibody gene fragments to form new antibody genes and clonal selection to pick only those newly made genes that make proteins that bind to the pathogen. DNA sequence analysis indicates the enzymes that perform the gene rearrangements are related by amino acid sequence to a family of enzymes called transposases found today in primitive chordates, the closest living relatives of the vertebrates.<sup>8</sup> Transposases cut short segments out of DNA and splice them into new locations in the DNA molecule. The DNA moved is called a transposon. When a transposon includes the gene sequence encoding the transposase responsible for moving it, the unit (transposon and produced transposase) can be thought of as a DNA parasite that reproduces by splicing itself into new locations in the host DNA. Data suggest that somehow one of these mutated so that the transposase it encoded joined pieces of antibody-coding DNA instead of its own DNA. Refinement of this initial error opened up the possibility of producing a lot of different antibody sequences without loading up the DNA with millions of antibody genes. The scenario only makes sense if there was already a mechanism in place for selecting and amplifying cells that produced antibodies that bound to foreign antigens. It is likely that before the transposase gene mutation, primitive chordates had multiple antibody genes and a mechanism for identifying and amplifying whichever one best matched an intruder.

The general scheme whereby cells divide and some are selected to divide further is as old as multicellular life and much older than the immune system. Selection favoring some cells over others during development is determined by cell surface proteins binding to other

proteins (or to polysaccharides). Cells are eliminated when the wrong proteins bind or the right proteins don't bind. Clonal selection in the immune system is similar except selection and subsequent stimulation to undergo rapid cell division is triggered by binding to an environmental protein (or polysaccharide) that is not part of the body's normal repertoire of (self) macromolecules. A simple change from stimulation by molecules that are a normal aspect of the body to molecules that are not is the likely first step that eventually led to adaptive immunity.

If this scenario is correct, selection preceded the initiation of a full-fledged engagement of the engine of complexity within the immune system. Creation of adaptive immunity was relatively easy when compared to the origin of life because a self-replicating information-encoding system was already in place and because key features of what would become the adaptive immune system already existed for other purposes. Key elements of the modern immune system are cell division (with DNA encoding of proteins), antibodies, T-cell receptors, a recombination system for rapidly generating binding protein variation, and selection based on cell surface proteins that bind specific molecules. Cells and proteins existed before any creature had an immune system, and as we have seen, all multicellular organisms employ cell lineage selection based on what cell membrane proteins bind to in their environments.

## BRAIN DEVELOPMENT

As discussed briefly in chapter 7, the multicellular body is constructed, in part, by mechanisms that select particular lineages of dividing cells over others. Elimination of some cells and proliferation of others builds new structure. In the brain this selection-based strategy extends to the subcellular level and also determines the detailed contacts made between individual nerve cells. The minute details of brain wiring are created by a simple strategy of favoring some randomly made neuronal connections and rejecting others. As neural processes grow outward toward broadly defined distant targets, they pass over cellular surfaces, each with specific molecular signatures. Recognition at the molecular level leads variously to continued growth in the same direction, synapse formation, or retraction and subsequent exploration in other directions. Exploration and selection when combined tap the inherent creativity of randomness (in the form of which way to go next) without engaging

the full power of the engine of complexity. In brain development there are no instructions specifying connection details that are copied, modified, and improved. There is no body of information that specifies the detailed pattern that forms. It is true that developmental systems are characterized by the recording of past choices in their ongoing structure, but in brain development this information is not repeatedly copied and modified to achieve ongoing refinement. Brain wiring provides testimony to the power of repeated selection when channeled by clever selection rules. Those rules, of course, are encoded in an organism's DNA and were created over the long course of animal evolution.

In the ancient past of animal life there occurred a change in strategy from predetermined bodies and brains to structure formation based on channeled random trial and nonrandom selection. *Caenorhabditis elegans* (*C. elegans*) is a small nematode that is a favorite subject for study by scientists interested in cellular and molecular aspects of body development. This worm (only 1 mm in length) normally lives in temperate soils but also does well on petri dishes growing bacteria, its preferred food. The adult hermaphrodite worm (the typical form) has 959 cells that are always found (except in mutant worms) in the same body locations. Development is deterministic: if an embryonic cell is removed experimentally, the developing worm will be missing a body part that would have resulted from that cell. The nervous system is no exception. The adult worm has 302 neurons that occur in highly predictable locations and make highly predictable connections with one another.

*C. elegans* DNA encodes roughly 20,000 genes, hardly less than human DNA (estimated at about 22,000 genes). Obviously, humans are much more complicated, yet only a few more genes are required to make a human being than a simple worm. What does this tell us? I suggest the answer is that our body formation takes better advantage of the power of selection. In the human body, no cell locations are precisely predetermined. If an embryonic cell is lost, the progeny of other cells will take over the functions. Human bodies are more or less the same at the tissue and organ levels when people are compared but are quite different at the level of individual cells. This reflects a developmental strategy different from that employed by *C. elegans*. Evidently it takes 20,000 genes to construct an organism with 1,000 cells if the location and function of each of those cells is explicitly specified. The human body has 10,000,000,000,000 ( $10^{13}$ ) cells. Such complexity would be

utterly impossible if the precise location of every cell was predetermined by instructions in DNA. An ancient change in strategy from determinism to a variation/selection-based system is one of the things that make us possible. It is also what made complicated brains possible.

## THOUGHT

The origin of thought is poorly understood, but it is highly unlikely that *C. elegans* (a roundworm) has thoughts. Its 302 neurons do make decisions that improve the odds of obtaining food, mating, and avoiding hostile environments, but it stretches credibility to claim this simple activity constitutes thought. As far as we know, roundworm decisions are hard wired, something like an adaptive neural network (ANN) operating in a computer (chapter 6). Higher animals such as birds and mammals clearly do have thoughts. All one must do to become convinced of this is to observe a squirrel or a crow working to solve a problem such as how to get food from a “rodent-proof” bird feeder. Dogs can act in very thoughtful ways. We don’t really know, of course, what any animal is actually thinking because we can’t talk with them. Talking, after all, is how we discover what other people are thinking. I see no reason not to accept a straightforward interpretation of animal actions to mean they really are thinking when they look and act like they are thinking. They probably do not engage in the elaborate scenario-building that characterizes human thought, and they certainly don’t think in human language, but there is no scientific reason to question the proposition that birds and mammals have thoughts; certainly they learn. Learning can be easily documented even for invertebrate animals including insects and higher mollusks. Even the lowly roundworm with only 302 neurons can learn to approach or avoid tastes, odors, or temperatures associated with the presence or absence of food.

I speculated in chapter 10 that thought is possible because brain action incorporates the computational strategy formalized by the engine of complexity. Any theory of learning that evokes the engine of complexity operating in the brain on millisecond time scales requires that at some long ago time in the evolutionary history of animals a learning strategy was hit upon that involved generation in a primitive brain of multiple simultaneous interpretations of sensory input

accompanied by selection of one that best matched other sensory and remembered data. Once this happened, it seems a small step to engage in multiple rounds of modification and selection; the essence of the engine of complexity. This would be a natural and straightforward way to improve an imperfect internal world model.

Whatever the primitive world model encoded in *C. elegans* brains, it must have come about through the standard slow trial-and-error process of biological evolution. If you are small and live in the soil, your world is pretty predictable most of the time, but if you are larger and live in the ocean or on the land surface, you are regularly confronted with new situations. The ability to adapt in real (not evolutionary) time must provide enormous advantage. The first evolutionary steps leading to thought probably happened quite early in animal evolution. It is incredibly useful to have a mental model of one's local environment. Without it every movement is effectively random with respect to the environment. A good model allows one to know places to hide and where to go to find food. A mental model is necessary for even rudimentary thought. Asking mental questions involves placing oneself in different places or posing alternative actions. Should I look for food at site A or at site B? Should I hide in cranny X or cranny Y? Once this capability arose, an important function of the brain became choosing between alternative mental models.

The evolution of a thought and the generation of disease-specific antibodies differ from brain and body development in one very important way. Antibodies and thoughts are both based on copying and modification of previously encoded information. Unlike fixed body structures, transient connectivity patterns in the brain can be generated in multiple modified forms. The anatomy of the brain is highly repetitive, ideal for producing similar but slightly different connectivity patterns to be selected based on their fit to other active neural patterns.

Learning requires alternatives, and alternatives demand a mechanism for selecting one over another. When those selections are solidified, learning occurs. Because body responses (movements, memories, thoughts) are determined by neural activity patterns, and neural activity patterns encode information, learning even in simple animals involves selecting one information-encoding pattern over another. It is not much of a stretch to imagine brain redundancy is organized in such a way that initially made and remembered decisions can be modified and improved



through an internal mechanism of repeated modification and selection. Thus, the roots of thought are probably indistinguishable from the roots of learning. The simplest nervous systems in living animals are found in the Cnidaria: hydra, corals, and jellyfish. Demonstration of associative learning in this primitive group has not been successful. The group also does not have a central nervous system, though they do have nerve cells and can respond positively and negatively to stimuli; but they do not seem to be able to make associations with indirect cues. All animals with a brain seem capable of rudimentary learning.

The anatomical structural innovations that enabled primitive brains to learn are poorly understood, but whatever they were, they must have been prestaged by an information-encoding system based on spatial and temporal patterns of cell-to-cell communication. This encoding strategy is utterly different from the genetic one. A subsequent critical step was probably the formation of a central nervous system that could entertain choices. This probably happened early in the Cambrian explosion of animal diversity.

## SOCIAL EVOLUTION

Given that we encode and modify information in our brains, that we communicate this to others imperfectly, and that we remember selectively, social evolution is inevitable. When thoughts and concepts are passed from one individual to another, information is copied in the process. Because we pick and choose what to remember and what to ignore, and because we sometimes embellish what we hear and read, it is hard to imagine how the collective information a society encodes would not evolve. The real issue may be understanding why primitive societies evolve so slowly.

The origin of social evolution is indistinguishable from the origin of communication. Social evolution is not limited to humans: the ability to open cars is widespread among the bears of Yosemite National Park but absent from most wild bear populations. Young bears apparently learn tricks from older bears. Bands of chimps and orangutans are well documented to use distinctive tricks for obtaining food that other bands do not use. Elaborate social institutions such as science and religion became possible only with the evolutionary achievement of speech, a new form of information communication.

The information that underlies social evolution today is not limited to spoken words, but also includes written words, diagrams, movies, music, and binary codes interpreted by electronic devices; but always, to contribute to society, information must pass through a phase of human brain encoding in the form of neural connectivity patterns. Social evolution is an implementation of the engine of complexity that is based on human learning, which as we have seen is quite likely a brain-based manifestation of the engine of complexity. Language evolved because it benefited those who could use it well in direct ways such as more accurate descriptions of opportunities and dangers. But, once language became possible, people began sharing innovations, preferences, and beliefs. Of necessity individuals selected what to pass on and what to ignore. The benefits of this were so great that there must have been enormous selective advantage to those groups capable of the best communication. The origin of human social evolution is inseparable from the origin of language, a new mode of information encoding. The origin of language is poorly understood but clearly required new brain structures (the result of DNA evolution).

## EVOLUTIONARY ALGORITHMS

Evolutionary algorithms, those instances of the engine of complexity that operate in electronic computers, are possible because of human technological innovations, themselves an aspect of social evolution. The origin of evolutionary algorithms is well documented. In the 1940s pioneer computer programmers began consciously emulating selected aspects of biological evolution in computer code. It worked. DNA is not necessary, nor cells, nor any other of the specific details that characterize living organisms. All that is needed for a man-made system to evolve in a computer is encoded information, a mechanism for copying this information, the introduction of modest changes during the copying process, and a mechanism for selecting some, but not all, of the copied and modified information according to consistent criteria. Clearly, the origin of these computer algorithms is human ingenuity and imitation of biology; and, of course, computer algorithms are not possible without computers. Because evolutionary algorithms are preplanned, their origins present no great mystery. People invent them.

At a deeper level, computer implementations of the engine of complexity are made possible by computer languages—new ways that information can be encoded and used. So, while human social evolution is enabled by human language, evolutionary computer algorithms are enabled by computer languages.

Computers are an outgrowth of social evolution, which is an outgrowth of a nervous system capable of communicating, thinking, and learning. The thinking brain is itself made possible by implementation of the principles of random trial and selection in the context of a complex developmental system specified by DNA nucleotide sequences that are themselves the result of uncounted rounds of experimentation and selection at the whole organism level. Thus, the situation we find ourselves in is the outcome of nested implementations of the engine of complexity, each made possible by an older evolutionary system. The only system that did not (apparently) arise from an older evolutionary system is life itself.

Each example of the engine of complexity that we have discussed, with the exception of adaptive immunity, is characterized by a different way of encoding information that is used to create something or to make something happen—in other words, a new way to write instructions. This is shown briefly in table 13.1.

In each case, except the last, the engagement of the engine of complexity was preceded by a system that employed selection without any copying of encoded information, and in each case, except adaptive immunity, a new implementation of the engine of complexity was made possible by a new way of encoding and using information. Whether the very first implementation (the origin of life) was more difficult than later ones is not clear because we don't know enough about what happened. Clearly, though, all of the systems discussed except the origin of life were made possible by structures developed through the action of some already- operating engine of complexity. This observation raises the question of whether the future will yield new manifestations of the engine. If so, they will probably emerge from the activities of already-existing engines of complexity and involve new ways of encoding and using information as instructions. More specific predictions are difficult to make. Who, one hundred years ago, would have predicted that social (i.e., technological) evolution would beget evolutionary computer algorithms that could solve hard engineering problems? Already

**Table 13.1**

Summary of five evolutionary systems.

Evolutionary system	Nature of information encoded	Form of encoding
Life	Catalysis and regulation of biochemical reactions and specificity of interactions between macromolecules	DNA and RNA sequences built of four monomer units
Adaptive immunity	Protein structure that binds to specific microorganisms	DNA nucleotide sequences
Learning and thought	Representations of the external world based on sensory input and potential future actions in that world	Temporal and spatial communication patterns among nerve cells
Society	Things learned from others and consequences of human behavior	Patterns of speech (human language)
Evolutionary algorithms	Computer actions and outcomes	Computer code

evolutionary algorithms are being put to work designing more powerful computers. The implications of this are remarkable and will be explored in the next chapter.

### What does it mean to be in the middle of a computation?

One natural consequence of the computational viewpoint is that you, your body, your thoughts, your accomplishments, and your interactions with others are current states of ongoing computations. If this seems a little unsettling, it is because you have not yet fully accepted the notion that computation underlies everything. The separation of computation from physical and biological processes is an historical accident of the way scientific understanding developed over the past four-hundred years. In nature, there is no such separation.

Since most complicated things are too improbable to exist without instructions to specify their details, and most things we care about are

complicated, most things in our worlds depend on instructions for their existence. The engine of complexity is a simple yet remarkable computational strategy that allows the improvement of instructions when nothing is known in advance about what changes would constitute improvements. Repeated selection of blind trials works. Recording those selections in some encoding medium allows repeated acts of selection to accumulate and add up to something. By this means the engine of complexity makes possible and also explains the existence of all things, including ourselves, that are too complicated to otherwise exist. Depending on details of each system, including the environment it operates in, the engine of complexity can continue innovating, seemingly without end. Two obvious examples of this are life, where the computation has been going on for 3.5 billion years, and human social evolution, which has been ongoing for perhaps two hundred thousand years. In other implementations, the engine reaches a goal and stops such as with adaptive immunity and most evolutionary algorithms. Human thought is a situation where the computation proceeds for the life of an individual (assuming that thought does indeed depend on the engine of complexity).

We are accustomed to thinking of computations as processes that manipulate information according to preset rules and then stop yielding defined output, say the result of a calculation or a document printed. Engine-of-complexity computations also produce outputs, but those outputs are used as inputs to another round of computation. The cyclic process can go on indefinitely, stopping only when a preset goal is reached or the physical basis of the computation is disrupted. A comforting feature of being an intermediate stage in an ongoing open-ended probabilistic computation is that even though the future is unknown, no possibility is ultimately out of reach.

---

## The Future

### Where is it all going?

Futurists have a terrible record of predicting the future. There is a good reason for this. A central feature of the engine of complexity is that trajectories of outcomes through possibility space are chaotic in the sense that there is no way to predict in detail where they will go. This reflects the underlying source of creativity, random change; still, predicting the future is a fun thing to do. When a system is characterized by an optimum, engine-of-complexity-based changes will home in on that perfect solution; but even in those cases, paths to the predictable outcome are unpredictable. Think back to the one-max problem discussed in chapter 6: the program stops when the output is all 1s, but there are lots of ways to get to that answer. Selection acts on real choices, and the ones offered up are all that is available. Random change presents possibilities, selection says yes or no, and the future is built on whatever is chosen. When there is no optimal solution, such as with coevolution, outcomes can go on changing forever. Always the path is unpredictable if the underlying variation is random.

Sometimes though, general trends are discernible. Convergent evolution illustrates this in biology. We can be pretty certain that in a slowly drying climate plants will evolve thorns and small leaves, and that in a slowly cooling climate, mammals will evolve thicker fur. A more controversial possibility is that under certain conditions, evolutionary systems predictably generate increasingly complex outcomes. In coevolutionary systems the rules of selection evolve as the populations evolve because each member of a community creates aspects of the environment for others. As discussed in chapter 12, when rapid but computationally demanding responses are advantageous we should expect depth, an aspect of complexity, to increase. An evolutionary situation we are constantly reminded of is technological development, where past advancements provide the foundation on which new advances are made. When change builds on previous change the effect can be dramatic. An example of this is Moore's law,<sup>1</sup> which describes the exponential growth in computer capability over the past half century.

Ray Kurzweil,<sup>2</sup> computer scientist, futurist, and popular writer, speaks and writes about the long-term implications of exponential increases in technological capability and exponential decreases in cost. He focuses on three technologies: computer power, biotechnology, and nanotechnology, which he sees merging in the not-too-distant future. Some of his projections are hard to argue with, even though they are hard to believe. We will consider each of these in turn.

## COMPUTER POWER

Gordon Moore was a pioneer in the development of integrated circuits and cofounder of the Intel Corporation. In 1965 he famously predicted that the number of transistors that could be placed in an integrated circuit would double every two years for decades to come. Figure 14.1 plots the logarithm of the number of transistors in various commercial microprocessors plotted against the year of introduction. There is a remarkable fit to the prediction. Other experts now project that with current technologies the trend is likely to continue for another decade. This would imply another twentyfold increase.

Kurzweil points out that Moore's law characterizes the most recent leg of a much longer trend that includes five distinct computing technologies;

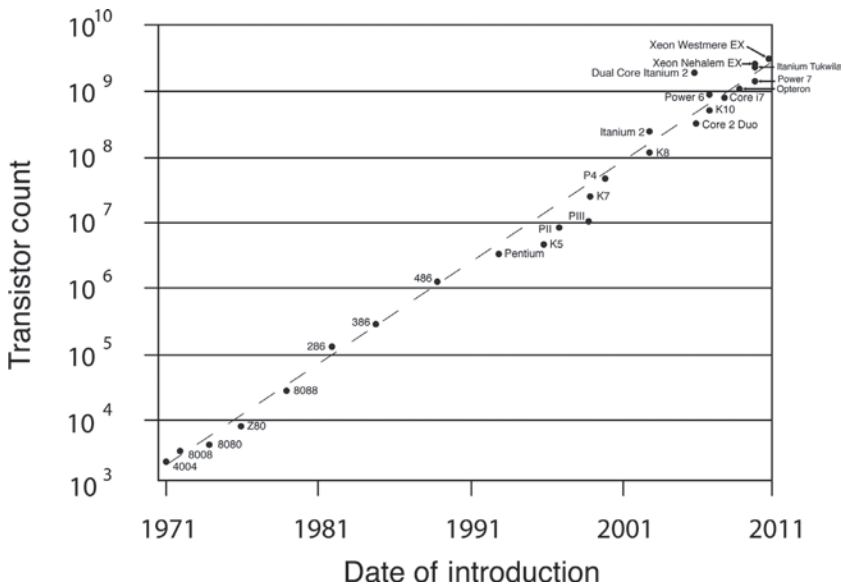


Figure 14.1. CPU transistor counts, 1971–2011. The logarithm of the number of transistors per integrated circuit is plotted against the date of introduction of various commercially successful chips. The dotted line shows the two-year doubling projection, starting with 2,000 transistors per chip in 1971. Data is from the Wikipedia entry on Moore’s law.

he calls them paradigms. These are: electromechanical devices, electronic relays, vacuum tubes, transistors, and integrated circuits. Introduction of each new paradigm resulted in smaller, faster, cheaper devices. Figure 14.2 plots calculations per second per thousand dollars of commercial computing device for all five paradigms since 1900.

Kurzweil predicts that the current paradigm, integrated circuits, will be replaced before 2020 by a new paradigm, possibly molecular computing, that will extend the trend further into the future. He explains it is not uncommon for technologies to develop and mature and then be replaced by different technologies. Each new technology, he observes, is characterized by three phases:

1. Slow growth (early phase of exponential growth).
2. Rapid growth (late, explosive phase of exponential growth).
3. A leveling off as the paradigm matures.



## Moore's law

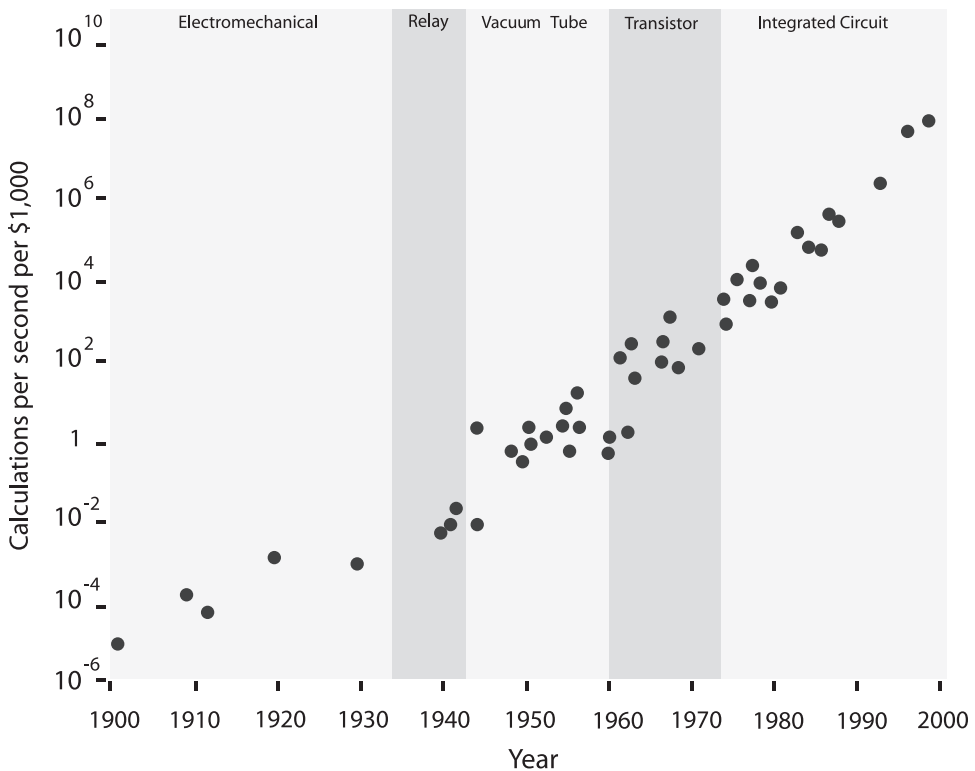
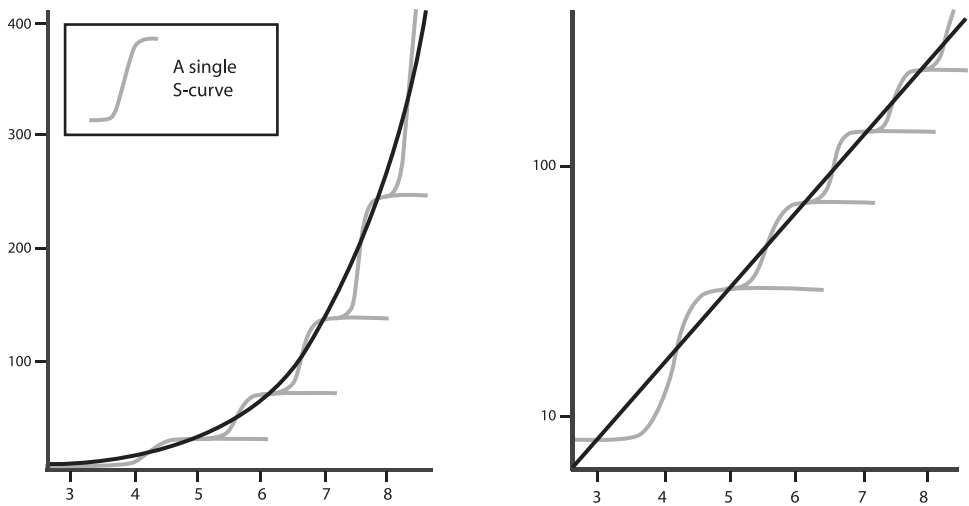


Figure 14.2. The logarithm of calculations/second/\$1,000 of various commercial computing devices since 1900. These devices were based on five distinct technologies.

Following an initial breakthrough, or invention, the development of any new technology proceeds slowly at first and then increases in speed as refinements are made and more engineers apply themselves to adapting the idea to new situations. This phase is also characterized by rapidly decreasing cost. Eventually, diminishing returns set in where improvements are harder and harder to come by, and the technology undergoes increasingly minor changes. This is the mature phase. A few examples of mature phase technologies are kitchen refrigerators, microwave ovens, automobile carburetors, and lead-acid batteries. If a technology is such that one can measure its capability, a plot of capability



*Figure 14.3.* Technology progress seen as built on a succession of technologies each characterized by the three phases of S-curve development. Left panel is performance plotted against time, and the right panel is the logarithm of performance plotted against time showing a straight line similar to that seen in figure 14.3.

versus time will be S shaped. As each technology matures, it is often replaced by a new technology whose development is itself characterized by an S-shaped curve. Chains of technologies are thus possible. The idea is shown in Figure 14.3 (left panel); and when the logarithm of capability is plotted against time, the result is a (nearly) straight line sloping upward (right panel). The waviness is not apparent in most measurements because real data has a lot of scatter. If instead of capability the cost per unit of capability is plotted, the characteristic result is similar but with a downward trend.

An electronic computational technology that is now on the horizon is imitation of the human brain. The brain has a logical structure that is totally different from your desktop computer and does some things better. The recent parallelization of fast computers may be a step in this direction since brain architecture is massively parallel. Brains are creative, very good at pattern recognition, and deal easily with logical inconsistencies, both weaknesses of digital computers. For this potential to be realized, the computational logic of the human brain will need to be better understood.

## BIOTECHNOLOGY

The reality of Moore's law is undeniable, but is it just an oddity, or are long-term exponential advances also observed in other technologies? Two that Kurzweil focuses on are biotechnology and nanotechnology. Biotechnology comprises a suite of capabilities concerned mostly with changing and moving DNA from one organism to another. Central to these technologies is the capacity to accurately and inexpensively determine the sequences of DNA molecules, and DNA sequencing is currently in a phase of remarkably rapid development. Techniques for DNA sequencing were first invented in the early 1970s, and they were labor intensive and expensive. By 1990 chemical innovations and improvements in automation made feasible the announcement of the human genome project. This multiagency government-sponsored project established a goal of determining the sequence of all 3 billion base pairs contained in one human DNA within fifteen years. At the time of the announcement the cost of determining a sequence was about \$10 per base pair. It was projected that investment in technology improvements would make sequencing faster and cheaper. In keeping with this, the cost goal for the project was set at 3 billion dollars, about \$1 per base pair, a tenfold improvement. The project was completed three years ahead of schedule for an estimated 2.7 billion dollars. Because the investment in technology was so successful, in the year 2000 another group headed by Craig Venter was able to use newly developed technology to sequence a second human genome in only nine months at a cost of 100 million dollars (3 cents a base pair). In 2007 James Watson (codiscoverer of the structure of DNA) announced that he had obtained the sequence of his own DNA. The company that performed the work reported it cost 1 million dollars and took two months to complete. Thus, in 2007 the cost of sequencing (in a large project) had dropped to 0.03 cents per base pair. In 2008 the National Institutes of Health announced the 1000 Genomes Project with the goal of sequencing 1,000 genomes for under \$50,000 each. On October 26, 2011, the X PRIZE Foundation announced major changes in its Archon Genomics X PRIZE. The \$10 million prize will be given to the first group to sequence 100 human centenarian genomes in thirty days or less for no more than \$1,000 per genome with an error rate no greater than one error per million base pairs. The competition was

set to begin on January 3, 2013. Each competing group is provided with the DNA samples at the start of the competition. If the prize is awarded and the goal of \$1,000 to sequence a human genome is achieved, which seems likely, then over a span of twenty years the cost of sequencing one person's DNA will have dropped from \$3 billion to \$1,000 and the time required from fifteen years to seven hours. Clearly, DNA sequencing is in the explosive phase of technology improvement.

These advances are based on five distinct lines of technology improvement: chemistry, automation, parallelization, miniaturization, and improvement in computational techniques for analyzing the massive amounts of data generated. Automation and parallelization depend on miniaturization, and all but chemical advancements depend on computer control and analysis. This demonstrates how technology improvements tend to benefit from and build upon progress made in other fields.

Nonsequencing aspects of biotechnology are not so easy to measure, but there is little doubt that capability is increasing at a remarkable rate on many fronts. Explosive progress is enabled in part by the increasing availability of inexpensive DNA sequence data and by increasing availability of powerful computational techniques.

What are the future implications? One is personalized medicine. The reason the National Institutes of Health is spending taxpayer money to bring the commercial cost of sequencing down so low and so fast is to make it possible for every person's medical record to include the sequence of their DNA. This, in turn, will make it possible for medical prescriptions to be individually tailored to the genetic makeup of each person. Currently, medical intervention is hampered because people react differently to the same dose of a medicine. Medicines interact with proteins, and proteins in the body are determined by DNA sequences, so, if the sequences are known, data can be assembled relating drug responses to the proteins of each patient.

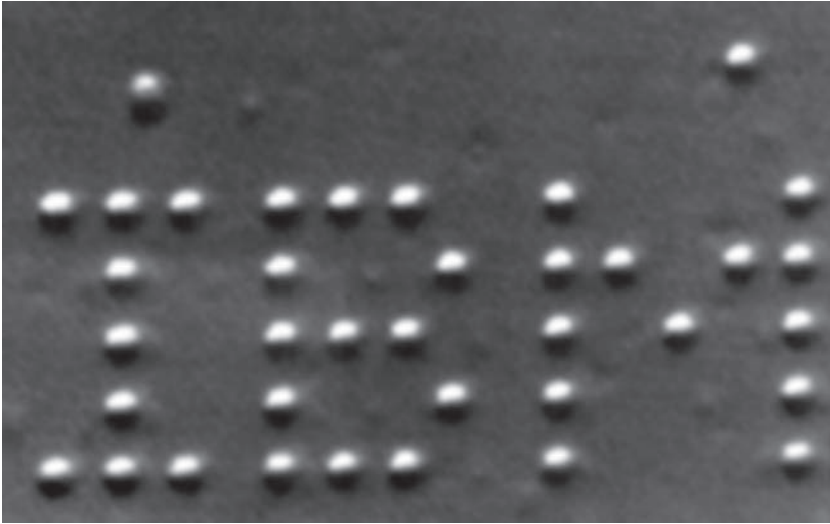
A very different, more distant, but still not so far-fetched idea is designer organisms. If the pace of biological discovery and parallel technology developments continues to accelerate, it is not much of a stretch to imagine a situation where a genetic engineer will sit down at a computer and "design" a new crop or a new microorganism for degrading oil spills, or whatever. We are on the verge of this capability now with viruses being redesigned to have useful medical properties.

The goal of virus redesigners is gene therapy. Some people are born with a defective gene. The idea is to fix it by inserting a normal gene into the body's cells. Viruses can be used to carry genes into body cells. The concept has already been tested in patients, and current refinements are likely to make the procedure routine within a decade.

## NANOTECHNOLOGY

The third leg of Kurzweil's technological vision of the future is nanotechnology. This is engineering focused on extreme miniaturization. Reduction in device size has been going on for a long time. Think of how industrial technologies of the late nineteenth century were dominated by gigantic machines with huge gears and enormous levers appropriate for doing mighty tasks. Some of these great machines are still functioning to raise bridges or close flood gates, but they have mostly been replaced by smaller, more powerful devices that accomplish the same jobs better. To give an idea of how far things have progressed, in 2003 a group at the University of California at Berkeley announced the creation of an operational motor that was 300 times smaller than the diameter of a human hair.<sup>3</sup>

Since the notion of a device is very general, it is difficult to create a graph showing the decline in minimum device sizes over time for devices in general. But, consider time pieces, clocks. Water clocks were invented by the Egyptians around 1400 B.C.E. The first were simply pottery bowls with a small hole in the bottom. By inscribing lines on the inside, the passage of time could be noted as the water ran out. Later versions had moving parts powered by flowing water. The spring-driven clock was invented around 1400 in Europe. The first models were not very accurate but could be moved from place to place without changing the plumbing. The first pocket watch was apparently made around 1462, with surviving examples from the early 1500s. The early models were pretty bulky by today's standards. The first wristwatch was produced in 1862, and by the late 1920s commercial watches could be made small enough to appeal to fashion-conscious women. The first electric clock was patented in 1909 by Henry Warren; these clocks were set in handsome pieces of furniture. In 2008 Epson Toyocom announced an accurate clock chip that measured only  $3.4 \times 1.7 \times 1$  mm. The chip, of course, does not have



*Figure 14.4.* The letters *IBM* spelled out in xenon atoms on a smooth surface of nickel metal (Copyright, Nature Publishing Group).

a clockface, otherwise you would need a magnifying glass to read it. Over the history of technology, clocks have gotten steadily smaller, cheaper, and more accurate. Stories similar to this can be found for most technologies.

A device we previously talked about is the transistor. Reduction in transistor size underlies Moore's law. Ultimately, the simple fact that material things are made of atoms must limit miniaturization. An object cannot be smaller than a single atom, and a device such as a transistor or a motor or a clock that has multiple parts cannot be smaller than one atom per part. The last frontier in miniaturization will be the ability to manipulate individual atoms to do what we want. A striking step in this direction was taken in 1990 when scientists at IBM Corporation demonstrated they could position individual atoms on a flat surface using an atomic force microscope. Figure 14.4 shows the letters *IBM* spelled out with thirty-five atoms.<sup>4</sup> It is a long way from positioning xenon atoms to building an atomic dimension motor or a nanobot that repairs molecular-scale damage by positioning individual atoms, but a number of far-thinking scientists and engineers see that capability as their ultimate goal.

Many researchers in this area are looking to biology for ideas and techniques. A host of very small machines including motors, ratchets, pumps, light sensors, tiny tram systems, and even clocks already exist and function in living cells. They are all very small. A suite of strategies for making small molecular devices is to modify, copy, or control things life is already doing. So, it seems that aspects of nanotechnology will merge with biotechnology. Will we live to see the day when nano devices engineered to identify and destroy cancer cells are injected into a patient's blood stream? Perhaps.

Making small things is computationally demanding because small things that behave the way we want demand great assembly precision and because the behavior of atoms and molecules is not easy to predict and control. So, it is easy to imagine a cycle of mutual dependency in which advances in biotechnology and nanotechnology depend on increasing computational capability and increasing computational capacity depends on nanotechnology and biotechnology.

A concern of many who marvel at the pace of current technological advancement is that our machines will replace us. Kurzweil offers a different vision. He sees us *merging with our technology*! Our brains, after all, are information processing devices. They are biological computers, albeit with a totally different logical structure than our desktop machines. Three early indicators of what is to come are the number of graduate programs that have sprung up in the area of "human computer interaction" (from none in 2000 to sixty-nine in the United States in 2009), the number of new books offered by Amazon.com in 2007 and 2008 on the subject of brain-computer interaction (six), and the number of popular science articles and news pieces on the subject (many). To quote from a CBS special that aired November 2, 2008, Scott Pelley reporting:

Quietly in a number of laboratories, an astounding technology is developing that directly connects the human brain to a computer. It's like a sudden leap in human evolution—a leap that could one day help paralyzed people to walk again and amputees to move bionic limbs.

The story features Scott Mackler, who suffers from amyotrophic lateral sclerosis (ALS), a neurodegenerative disease that has left him totally paralyzed except for eye movement. By wearing a removable

helmet-like prosthesis, Scott was able to participate in the television interview by thinking of what he wanted to say and having his words spoken by a computer. Current technology is slow, and it took Scott about twenty seconds per letter to respond to a question. But even so, he could communicate, and communicate surprisingly well. The first commercial brain-computer interface designed for the computer gaming market was announced in February 2008 by Emotiv Systems. It is a headset with 14 sensors that allows the wearer to control personal computer actions by thinking of them. The retail price is \$299. This remains a novelty item, since for most people their hands and arms are much better, but what will the technology be like in ten or twenty years?

A more developed technology is the cochlear implant. This commercially available device worn by over one-hundred and fifty thousand people in 2009 allows profoundly deaf people to hear. It works by connecting a small microphone to an electronic speech processor (computer) that sends electric impulses to an array of electrodes implanted in the auditory nerve. The brain then converts these impulses into a perception of sound. The implant substitutes for the normal machinery of the inner ear and allows the wearer to hear, although current models are not as good as the healthy ear.<sup>5</sup>

So far, brain-computer interfaces are less capable than normal body functions, but the technologies are less than fifteen years old and rapidly improving. Imagine if you could wear a device that allowed your brain to access the speed and accuracy of a digital computer and yet still allowed you to be human? The idea is not so far fetched.

Kurzweil's professional expertise is computer technology. He observes that if the current exponential increases in computational power and exponential decreases in cost continue for only forty more years, the world we live in and the very nature of humanity will totally change. If the apparent limit to information propagation set by the speed of light can be circumvented, then in only two hundred years the entire universe will be transformed. This is pure science fiction, but in the relatively near future we can look for two key transitions. The first will be the creation of computers that design the next generation of computers. This will allow technology to escape from the limitations of human thought. Already, computer chips are too complicated for the designing engineer to fully comprehend; software plays a significant



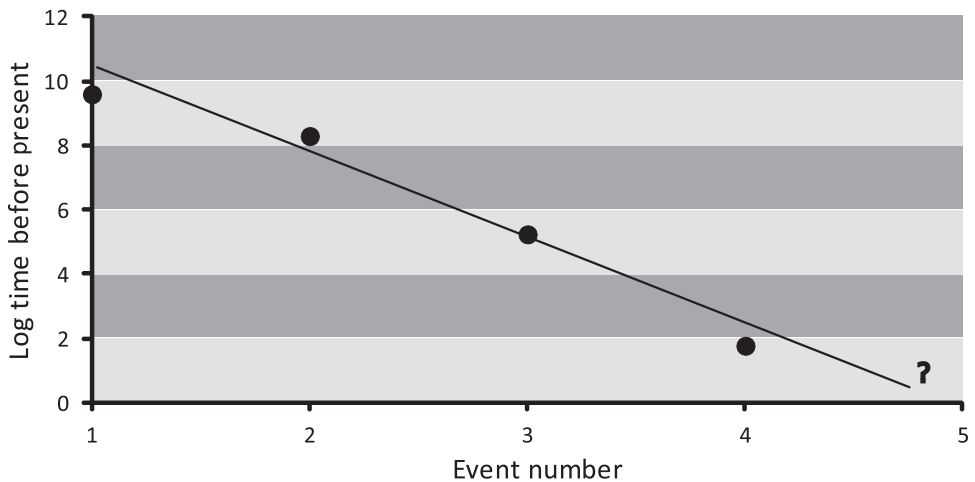
role in the design of each new generation of Intel chips. How much longer will it be until the engineer is cut out of the loop? The second is for computers to have the capability of simulating the human thought process. This will allow computers and people to interact electronically.

If a brain-computer interface could be achieved in which large amounts of bit-level information were exchanged between person and machine, then human thought could escape the confines and limitations of the human body. Because information can be copied it need not be confined to any particular physical system. If our thoughts could be copied to a machine capable of human thought, our thoughts would not die with our bodies, and we (our memories and thoughts) would become immortal. Kurzweil projects this happening in the 2040s and is doing everything possible to assure that he lives until that time. It is less clear if his vision involves him living on in his current body, but I am sure, if I asked him, he wouldn't care!

### Are we about to participate in a new implementation of the engine of complexity?

When new implementations of the engine of complexity are considered over the history of our planet, two patterns emerge. First, systems that implement the engine seem to spawn new implementations (discussed in chapter 13), and second, the time between each new implementation gets shorter. The origin of life somehow underlies all the other implementations. We know of no examples of the engine of complexity that are not associated with life. Figure 14.5 shows the trend. I omitted the adaptive immune system from this figure because it seems to be a side shoot, not likely to generate a future change central to the future of humankind, and included animal thought, even though a central role for the engine of complexity is not confirmed.

The scenario goes like this: Life (somehow) started up about 4 billion years ago (event 1); animal brains capable of rudimentary thought first appeared roughly 200 million years ago (event 2); human speech and societies based on speech communication developed about 200,000 years ago (event 3); and computer-based evolutionary algorithms were developed only sixty years ago (event 4). Figure 14.5 plots the time of each origin on a logarithmic scale. There are only four data points, but



*Figure 14.5.* A plot of the logarithm of time before present versus event number. The numbered events are: (1) the origin of life, (2) the origin of thought, (3) the origin of speech-based human societies, (4) the origin of evolutionary algorithms, and (5) the present.

the trend is striking, and if it continues, another origin is predicted to occur any time now! Is the next step computers designing computers? Or computers whose architectural logic includes the engine of complexity? Or is it the merger of human thought with computer hardware? Or all of these? Or something else? Only time will tell, but if we don't annihilate ourselves, it seems likely the engine of complexity has some major surprises to offer up in the not-very-distant future.<sup>6</sup> These are exciting times. Stay tuned.



---

## Acknowledgments

First and foremost I thank my colleagues Dan Ashlock and Jack Lutz without whose continued interest and patience I would never have delved as deeply into the nuances of evolutionary computation and theoretical computer science. Without their many stimulating discussions this book would never have been written. My number-one fan was and is Matt Patitz, then an Iowa State University graduate student, and now an assistant professor of computer science and computer engineering at the University of Arkansas, who read and commented on early drafts and never failed to remind me how important my story was. Numerous other colleagues, including Leigh Testfatsion, Alan Atherly, Jim Lathrop, James Morrow, Ethan Scott, Don Sakaguchi, Doug Finnemore, Steve Kawaler, and Gary Howard, provided critical comments and suggestions along the way. Two department chairs, Duane Enger and Martin Spalding provided me with time to pursue my ideas. Several dozen undergraduate students read and commented on all or parts of the manuscript. You know who you are; I hope you enjoyed the experience and I want you to know that your efforts were greatly valued. Students were particularly helpful in identifying passages

that were not clear. Many of their suggestions are incorporated into the text. The ideas explored in several of the chapters were first vetted in a class I offered on complex adaptive systems. Most of the illustrations were professionally created by seven students majoring in biological and premedical illustration at Iowa State University. They are: Amy D'Camp, Audrey Gifford, Meaghan Hetherton, Brandon Holt, Tasha Obrin, Meg Pieperbrink, and Sean Siberski. Funds to pay these talented students for their time and expertise were supplied by a grant from the Iowa State University Publication Endowment Fund. Next, I would like to thank my agent, Peter Prescott, for his many useful edits and for leading me through the intricacies of publishing; and thanks also to Patrick Fitzgerald and Bridget Flannery-McCoy at Columbia University Press who turned my manuscript into a book. Finally, I thank my wife, Betsy, my children, Kate and Margie, and their husbands, Gordon Bowen and Ethan Scott, for providing encouragement, suggestions, and advice over the surprisingly long time it took to write this book. Of particular importance was their repeated reminders that the concepts I was presenting are not as simple as I would like to think; this is especially true of Betsy and Kate, who are themselves accomplished writers.

---

# Notes

## Introduction

1. Michael Ruse in *Darwin and Design* (2003) provides a brief, easily approachable history of philosophical thinking about purposeful complexity.

2. *Information and the Nature of Reality* (2010), edited by Paul Davies and Niels Henrik Gregersen, is a compendium of articles written by a wonderful collection of knowledgeable authors. It is an excellent source for up-to-date synopses of current philosophical discussions about information ranging from physics to religion. Seth Lloyd's chapter "The Computational Universe" summarizes some of the main points made in his earlier book, *Programming the Universe* (2006).

3. I consider *The Blind Watchmaker* (1986) by Richard Dawkins to be the best single explanation of biological evolution available for the nontechnical reader, though the computer simulations are a bit dated. Among college freshmen, the better students love the book, while the weaker students find it a bit over their heads. For those students, some of Dawkins's later books, such as *Climbing Mount Improbable* and *The Ancestors Tale: A Pilgrimage to the Dawn of Evolution*, are easier to read.

4. I make this statement notwithstanding a new book, *Biology's First Law, The Tendency for Diversity & Complexity to Increase in Evolutionary Systems* (2010) by

Daniel W. McShea and Robert N. Brandon. These authors consider ramifications of considering in isolation just half of the standard Darwinian explanation: generation of heritable diversity followed by selection. If one leaves out selection, then the authors elevate to the status of a “law” the natural consequence of diversity generation to be increasing diversity. Others have previously attempted to do the same with selection, proposing a “law of natural selection.” It remains to be seen if either of these attempts has the staying power to find their way into textbooks on evolution.

5. The reasoning here is that “laws” tend to be simple statements or mathematical equations. Evolution is a process, the input to an evolutionary computation is typically long and complicated, and the resulting outcomes are also complicated. Simple statements or relationships are rarely applicable in such situations.

6. After Dawkins, the second author I would recommend to any reader interested in evolution is Daniel Dennett. Dennett’s *Darwin’s Dangerous Idea* (1995) is probably the book closest to this book. He is a first-rate philosopher who understands evolution very well and, unlike so many philosophers, can write in a manner that is easily mastered.

7. Thomas Kuhn (*The Structure of Scientific Revolutions*, 1962) is generally credited with establishing “paradigm change” as a fundamental feature of scientific advancement. Kuhn and Karl Popper were the two most influential writers on the philosophy of science in the twentieth century. Popper argued for the evolutionary nature of scientific progress.

8. Steven J. Gould makes this point forcibly in *Wonderful Life: The Burgess Shale and the Nature of History* (1989). A wonderfully entertaining and insightful book about creatures that lived in a time often called the “Cambrian explosion.” This is when obviously animal fossils first appeared in the geological record.

9. Seth Lloyd is a theoretical physicist with strong ties to the Santa Fe Institute. His recent book, *Programming the Universe: A Quantum Computer Scientist Takes on the Cosmos* (2006), makes a case for the central role of information in cosmology.

10. Any of Steven Pinker’s books are well worth reading if one is interested in psychology and the brain. *How The Mind Works* (1997) is my favorite. Pinker argues persuasively for the idea that the brain is an information-processing organ and that behaviors we observe are consequences of that activity.

11. Social evolution is such a vast and still controversial field it is hard to know where to point interested readers. I suggest three books: Dawkins’s *The Selfish Gene* (1976), David Sloan Wilson’s *Darwin’s Cathedral* (2002), and Eric Beinhocker’s *The Origin of Wealth* (2006). Each is quite different: Dawkins writes on genetics, Sloan Wilson on religion (from a population biologist’s perspective), and Beinhocker on economics. Taken together they paint a powerful picture of the pervasive role of evolutionary activity in all social institutions.

## 1. The Problem

1. This specific recipe was taken from the HGTV website: [http://www.hgtv.com/hgtv/ah\\_recipes\\_poultry\\_eggs/article/0,1801,HGTV\\_3189\\_1394919,00.html](http://www.hgtv.com/hgtv/ah_recipes_poultry_eggs/article/0,1801,HGTV_3189_1394919,00.html). A quick Internet search for “easy soufflé” will reveal dozens of similar recipes.

2. Great Principles of Computing: <http://cs.gmu.edu/cne/pjd/GP>. This website is a terrific place to start if one is interested in acquiring a deeper perspective of computation.

3. *Meaning* is a slippery term in that is not well captured in information science. Partly this is because meaning is relative. It is context dependent. One can talk quantitatively about how much one information body correlates with another or how much change is caused in another system, or in an evolutionary system, how much information has been acquired about the environment of the system (Adami et al. 2000); but one cannot quantify how much a certain amount of information means to someone or something else.

4. The following equation allows one to calculate the amount of Shannon entropy characterizing a system:  $\mathcal{H} = -\sum p_i \log_2(p_i)$

The symbol  $\Sigma$  indicates a summation over  $n$  possible states,  $p_i$  is the probability of the  $i$ th state, and  $\log_2$  means logarithm to the base 2.  $\mathcal{H}$  has units of bits.

Use of the equation is straightforward. For a system with two possible states, the equation becomes:  $\mathcal{H} = -[p_1 \log_2(p_1) + p_2 \log_2(p_2)]$ , where  $p_1$  is the probability of state one and  $p_2$  the probability of state two. If the probabilities of the states are equal, then  $p_1 = 1/2$ ,  $p_2 = 1/2$ ,  $\log_2(1/2) + \log_2(1/2) = -1$ , and  $\mathcal{H} = 1$  bit. A system capable of only two states carries one bit of information if the probabilities of the two states are the same. If the probabilities are not all the same, the value of  $\mathcal{H}$  is less. Shannon formally referred to  $\mathcal{H}$  as entropy, but he and many others have considered  $\mathcal{H}$  to measure information.  $\mathcal{H}$  is maximal for a system when all the state probabilities ( $p_i$  terms) are the same.

5. A simple mathematical way to see why the information carried by an object (regardless of meaning) is equal to the entropy of the system of which the object represents one particular state is to define the information carried by a object to be the difference between the entropy of the object and the entropy of the system. Since the entropy of a single object is zero (its probability is 1, and the logarithm of 1 is zero), the information characterizing the object is the same as the entropy of the system.

6. One wishing to explore these concepts in greater depth must be prepared to study and understand the mathematics. A widely used text is *Elements of Information Theory* by Cover and Thomas (1991).

7. The idea here is to extend the logic first presented in note 5. The maximal entropy of a system occurs when all the probabilities in the Shannon equation (note 4) are equal. This is the case if the bits of a message are randomly chosen.



If the probabilities are not equal, then in some way the sequences are not random, and nonrandomness is the hallmark of structure. Adami et al. (2000) defined *complexity* to be the difference between the actual entropy and the maximal entropy for an evolutionary system and used it to measure how much information the system had acquired about its environment. The idea is that if an evolving bit string is random, it carries no information about its environment, whereas if the probabilities are unequal, this must reflect acquired information. I don't think this is a good use of the term *complexity* and would have preferred the authors to have used *negentropy* in keeping with the thermodynamic meaning of this term.

8. Every book on evolution has such a list. They differ slightly in detail, but can always be expanded or reduced to these five elements.

9. *Climbing Mount Improbable* is Dawkins's sequel to *The Blind Watchmaker*. In this book he tries to make clearer some points that readers of the earlier book misinterpreted or failed to understand. In keeping with its purpose, it is somewhat easier reading than *The Blind Watchmaker*, but any reader who really wants to understand evolution better would benefit by reading both.

10. Dennis Bray, *Wetware, a Computer in Every Living Cell* (2009) explores the thesis that the actions of cellular life constitute computations.

11. Newton's law of gravity can be written in mathematical form thus:  

$$F = G \cdot m_1 \cdot m_2 / r^2.$$
 $F$  is the force experienced by two masses,  $m_1$  and  $m_2$  that are separated by a distance  $r$ , and  $G$  is a constant. Later, I will emphasize that this, and others like it, are simple relationships. If you are skeptical of this claim, contrast the law of gravitation with the recipe for a soufflé or a computer program with a thousand lines of code.

12. The writings of Dawkins, Dennett, Pinker, Edelman, and Beinhocker referenced elsewhere in this work all document this assertion. Another interesting read within the same general theme is *Darwin Among the Machines* (1997) by George B. Dyson.

13. Michael Behe, *Darwin's Black Box* (1996), is probably the best and most discussed work that attempts to make the argument for irreducible complexity. The problem with all such arguments is they are ultimately based on what I call lack of imagination. More common is to call it "argument from personal incredulity" (Dawkins, *The Blind Watchmaker*, p. 38). In other words, statements are made that "this just couldn't happen," with no further proof. Any Web search for Behe will reveal many articles explaining why his arguments are insubstantial. For an eloquent review of the book, I suggest one by Robert Dorit that appeared in the September-October 1997 issue of the *American Scientist*.

14. *Endless Forms Most Beautiful: the New Science of Evo Devo and the Making of the Animal Kingdom* (2005) by Sean Carroll. In my opinion, this book is the best introduction available for a nontechnical audience to the molecular logic that underlies the creation through evolution of animal body structures.

15. Such figures are called fractals. This particular example is known as a Julia set in honor of Gaston Julia who first explored their properties. The figure shown in figure 1.1 was produced by the Matefitness project, Genoa, Italy ([www.matefitness.it](http://www.matefitness.it)) and is reproduced here with permission.

16. Daniel Dennett explains what he means by the intentional stance particularly well in *Freedom Evolves* (2003).

17. Richard Dawkins in *The Blind Watchmaker* quotes William Paley directly and presents his argument for why Paley's argument, which much impressed Darwin, is now known to be flawed.

18. Gary Cziko, in *Without Miracles* (1995), makes the case that all examples of complex fit have their origin in selection. His conclusion follows directly from writings by Donald Campbell and Karl Popper, which he reviews.

## 2. Computation

1. Great Principles of Computing website: (<http://cs.gmu.edu/cne/pjd/GP>) is highly informative with a lot of good and thoughtful information about computation.

2. I highly recommend Daniel Dennett's *Darwin's Dangerous Idea* (1995) to anyone wanting to learn more about evolution. Dennett's field of study is philosophy.

3. Logic gates and circuit designs based on them are easily found on the Web. Search for "logic gates" and you will find many interesting sites that provide much more information than I have given here.

4. Stephen Wolfram, *A New Kind of Science* (2002). The book is over 1,000 pages long but has an illustration on almost every page and is not very difficult to read. Wolfram hopes that the ideas in his book will "come to pervade not only science and technology but also many areas of general thinking." Permission to reproduce figures 2.3–2.6 does not imply endorsement by Wolfram Research.

5. *ibid.*

6. Erik Winfree's home page is <http://www.dna.caltech.edu/~winfree/>. He and his coworkers have pioneered the mathematical and physical exploration of programming structure by means of DNA tiles.

7. Figures 2.7 and 2.9 are inspired by and figure 2.8 is reproduced directly from "Algorithmic Self-Assembly of DNA Sierpinski Triangles," Paul W. K. Rothmund, Nick Papadakis, and Erik Winfree, *PLOS Biology*, pp. 2041–2052, Vol. 2, December 2004. License is granted by Creative Commons License Deed (<http://creativecommons.org/licenses/by/2.5/>).

8. Jack Lutz is a professor of computer science at Iowa State University who regularly teaches a graduate class in tiling theory. This quote is from a class he taught in 2007.

### 3. Structure for Free

1. Brian Greene, *The Elegant Universe* (2003); Seth Lloyd, *Programming the Universe* (2006); Lee Smolin, *The Life of the Cosmos* (1997), and any of a number of books written by Paul Davies are good places to start if one is interested in the origin of the universe and attempts to create a “theory of everything.”

2. See Stephen Wolfram, *A New Kind of Science* (2002), for hundreds of examples.

3. This chapter’s title was inspired by the phrase “structure for free” coined by Stuart Kauffman in *At Home in the Universe* (1995). Kauffman was describing structure that forms spontaneously in nonequilibrium situations. The phrase seems apt for all structures not requiring instructions for their formation.

4. In the mathematical expression of Coulomb’s law,

$$F = -\frac{k \cdot q_1 \cdot q_2}{r^2},$$

$F$  is the force,  $k$  is a constant that determines the units of  $F$ ,  $q_1$  is the magnitude of charge one,  $q_2$  is the magnitude of charge two, and  $r$  is the distance between the charges. By convention, the  $q$ ’s are positive if the charge is positive and negative if the charge is negative. This means that  $F$  is positive (the charges attract) if one charge is positive and one is negative, and  $F$  is negative (the charges repel) if both charges are positive or both are negative.

5. The time-dependent Schrödinger equation:

$$i\hbar \frac{\partial \Psi}{\partial t} = -\frac{\hbar^2}{2m} \frac{\partial^2 \Psi}{\partial x^2} + V(x)\Psi(x, t) \equiv \tilde{H}\Psi(x, t)$$

is the fundamental equation of physics for describing quantum mechanical behavior. It describes how the wave function of a physical system changes over time.  $\Psi$  is the wave function and  $V(x)$  is the potential.  $H$  is a mathematical function known as the Hamiltonian operator. Solutions of this equation are allowed energy levels.

6. Newton’s law of gravity:  $F = g \cdot m_1 \cdot m_2 / r^2$ , where  $g$  is a constant that determines the units of  $F$ ,  $m_1$  and  $m_2$  are two masses, and  $r$  is the distance between them.

7. The driving forces responsible for Bénard cells and similar phenomena were long misunderstood. The history of how the central importance of surface tension was recognized can be found in *Bénard Cells and Taylor Vortices* (1993) by E. L. Koschmieder.

8. The power law formula is  $y = ax^k$ . This formula describes an exponential relationship. Such relationships are common in nature. Taking the logarithm of both sides of the equation yields  $\log(y) = k \cdot \log(x) + \log(a)$ . When  $\log(y)$  is plotted

versus  $\log(x)$ , the resulting points form a straight line with a slope of  $k$ . Thus power laws are revealed when data plotted on a log-log plot form a straight line.

9. *How Nature Works* (1996) by Per Bak explores many examples of power laws observed in nature and experiments and makes the case that self-organization is important for understanding a wide range of phenomena.

## 4. Purposeful Structure

1. I met Nathan Johnson when he was a graduate student working on a project using evolutionary algorithms to create improved designs for simple-to-construct woodburning cookstoves for poor people in Central America. He was an undergraduate student when he took Mechanical Engineering 270 in 2003.

2. William Paley's *Natural Theology – or Evidences of the Existence and Attributes of the Deity Collected from the Appearances of Nature* (1802) is the most famous exposition of the argument for design of life by a higher power and stimulated the title of Richard Dawkins's *The Blind Watchmaker*. Dawkins effectively explains how scientific discoveries since that time have invalidated Paley's arguments.

3. A number of molecular details are glossed over in this section. There is a different tRNA for each codon–amino acid pair, and special enzymes assure that the correct amino acid is first attached to each tRNA. A particular codon, AUG, starts the process on mRNA, and any of three codons stop it. Various “helper” proteins called translation factors are required for the process to work. Those interested in a more complete accounting should read the molecular genetics chapters of any college-level introductory biology textbook. The one I have used for many years is *Biology* by Neil Campbell and Jane Reese. The book is now in its 9th edition and in the most recent edition Jane Reese is the primary author.

4. For wonderful animations of T4 assembly and infection go to the website: [http://www.seyet.com/t4\\_academic.html](http://www.seyet.com/t4_academic.html). The drawings in this book are taken from a 2003 review article, “Structure and Morphogenesis of Bacteriophage T<sub>4</sub>” by P.G. Leiman, S. Kanamaru, V.V. Mesyanzhinov, F. Arisaka, and M.G. Rossmann in *Cellular and Molecular Life Sciences*, vol 60, pp. 2356–2370.

5. Any introductory text, such as *Biology*, by Campbell and Reese will provide a somewhat more detailed explanation. For a complete explanation one needs to refer to a more advanced text. Two that I recommend are *Neuroscience: Exploring the Brain* 3d ed. (2007) by Mark F. Bear, Barry W. Connors, and Michael A. Paradiso and *Neuroscience*, 5th ed. (2012), Dale Purvis ed.

6. *The Selfish Gene* (1976) was Richard Dawkins's first book and arguably has had more influence than any of his other books on the biological and social sciences. Two insights, in particular, have stood the test of time. First, bodies are devices for the propagation of genes, and, second, mental concepts shared within

a culture can be characterized in genetic terminology. In the book he coined the term *meme* to describe the social equivalent of a (DNA) gene.

## 5. Improbability and the Engine of Complexity

1. In *The Hitchhikers Guide to the Galaxy* (1979) Douglas Adams built from the many-worlds interpretation of quantum mechanics a humorous and fanciful account of intergalactic travel based loosely on the idea that future technology would allow travel between parallel universes.

2. “Numerical Cognition Without Words: Evidence from Amazonia.” An article by Peter Gordon in the October 15, 2004, issue of *Science* magazine presents clear evidence that some languages do not admit counting beyond two.

3. A number of good Internet sites describe in detail how to perform basic arithmetic on Roman numerals. Search for “Roman numeral arithmetic.”

4. Daniel Dennett, in *Darwin’s Dangerous Idea* (1995) uses Jorge Luis Borges’s 1962 image of a library of all possible books, the “Library of Babel,” presented in *Labyrinths*. Dennett goes on to create a subset of the Library of Babel that he calls the Library of Mendel, a library of all possible DNA sequences.

5. Small probability reference: Risk to groundlings of death due to airplane accidents: a risk communication tool. By Bernard Goldstein, et al., *Risk Analysis* 12 (1991): 339–341. Data from the United States, 1975–1985.

6. *On the Origin of Species by Means of Natural Selection* by Charles Darwin, 1859. This is possibly the most influential and widely quoted science book of all time. It is still quite readable.

7. *Endless Forms Most Beautiful*, 2005, by Sean Carroll is probably the best popular introduction to EvoDevo currently available.

## 6. Algorithmic Evolution

1. Computer learning or artificial intelligence as it is often called is a rather technical subject if one’s goal is to understand in detail how computers are able to learn. A good readable introduction is provided by *Understanding Artificial Intelligence*, a collection of *Scientific American* articles published in 2002.

2. John Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*, 2nd ed., 1992; David Goldberg, *Genetic Algorithms in Search Optimization and Machine Learning*, 1989. Neither of these books is particularly helpful to the nontechnical reader.

3. Tartarus is a computer game invented by Astro Teller, the grandson of Edward Teller, nuclear physicist who led the U.S. effort to develop the hydrogen

bomb, and Gerard Dedbreu, Noble laureate in economics. Teller describes himself as an entrepreneur, scientist, and author. He invented the game while a graduate student at Carnegie Mellon University as a means of exploring machine learning. Dan Ashlock picked up on the idea and further explored it with his students and devotes a chapter to Tartarus experiments in his 2006 book on evolutionary computation (Ashlock, 2006).

4. Goldberg, 1989; Ashlock, 2006.

5. All introductory biology texts introduce this visual aid, but the original credit is usually given to Sewell Wright, “The Roles of Mutation, Inbreeding, Crossbreeding, and Selection in Evolution.” In *Proceedings of the Sixth International Congress on Genetics*, pp. 355–366 (1932).

## 7. Evolution Within the Body

1. The topics of cell migration and programmed cell death during development are discussed in any modern developmental biology textbook, for example, Lewis Wolpert, et al., *Principles of Development*, 3rd ed., (2006) and Scott F. Gilbert, *Developmental Biology*, 8th ed., (2006).

2. Gerald Edelman won the Nobel Prize in Physiology or Medicine in 1972 for his work in immunology and then turned his attention to the nervous system. His book *Neural Darwinism* (1987) lays out his ideas about how variation and selection play a central role in brain processes. *A Universe of Consciousness* (2000) by Edelman and Giulio Tononi provides a more readable update of his ideas.

3. Recent introductory-level college biology texts typically devote a chapter to the mammalian immune system. Reading such a chapter is a good way to achieve a decent introduction to the subject. For further detail, intermediate-level texts devoted entirely to the subject provide an excellent overview, such as *Immunobiology* (2004) by Janeway, Shlomchik, Travers, and Walport.

4. In human medicine the term HLA (human leukocyte antigen) is often used instead of MHC (major histocompatibility).

## 8. Taking Control of the Cycle

1. Once again, any introductory biology text will provide an account of random assortment and recombination (sometimes called “crossing over”).

2. David Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning* (1989). Goldberg’s definition of a genetic algorithm is one that includes recombination.

3. Charles Darwin, *On the Origin of Species by Means of Natural Selection* (1859).

## 9. Complex Systems

1. Stuart Kauffman, *Origins of Order* (1993), *At Home in the Universe* (1995), *Reinventing the Sacred* (2008). Kauffman was a long-time member of the Santa Fe Institute and possibly the person most responsible for explaining to the noncomputational world the discovered principles of the new field of complexity science and the power of complex adaptive systems.

2. Figure 9.8 is adapted from a review article in the journal *Cell* by Cyran, et. al. published in 2003. The molecular details of the circadian clocks that run in every cell are not fully understood for any organism and differ significantly across the spectrum of life. On the Web, a good place to explore further is the NIH web-site (<http://www.nih.gov/>) and search the site for references to “biological clock.”

3. This particular list was taken directly from the Wikipedia entry on the subject that was active when I was writing this chapter. Two good, very readable books on the subject are by John Holland, *Hidden Order* (1995) and *Emergence* (1998).

## 10. Human Learning and Creativity

1. Eric Kandel, *In Search of Memory, the Emergence of a New Science on Mind* (2007). Kandel provides a lovely review of the modern science of the mind by recounting his successes and life as a neuroscientist.

2. Karl Popper, “Of Clocks and Clouds.” Chapter 6 in *Objective Knowledge: An evolutionary Approach* (1972). This chapter is the formal presentation of the Arthur Holly Compton memorial lecture presented at Washington University, April 21, 1965. This beautifully written lecture lays out the argument for the role of randomness and selection in human thought. It is also an excellent introduction to Popper’s work.

3. For a basic introduction to muscle function, the reader is again recommended to an introductory-level college biology text, such as *Biology* by Campbell and Reese (2008).

4. A good starting place for a greater understanding of chaos is *Chaos, Fractals, and Dynamics, Computer Experiments in Mathematics* by Robert L. Devaney (1990).

5. Arthur H. Compton, *The Human Meaning of Science*, pp. 45–46 (1940). With this example, Compton begins his argument that because of quantum mechanics the classical law of causality does not hold at the molecular level. From this Compton argues, to his great relief, that human free will is permitted by modern physics.

6. A central theme of philosophy from ancient Greek times to the present is human free will. Newton’s laws and most physical laws discovered prior to

quantum mechanics are formulated in terms of exact equations. Nearly all Western philosophers and most scientists came to interpret this to mean the universe is deterministic. Both Compton (*The Freedom of Man*, 1935; *The Human Meaning of Science*, 1940) and Popper ("Of Clocks and Clouds," 1972) persuasively argued that quantum mechanics put an end to such a worldview. Even so, the idea is still discussed as if it has philosophical and/or scientific substance.

7. Steven Pinker, *How the Mind Works* (1997) provides an excellent and highly readable introduction to brain function.

8. Donald T. Campbell, "Evolutionary Epistemology," Chapter 12 in *The Philosophy of Karl Popper*, P. A. Schlipp ed., pp. 412–463 (1974).

9. Gerald Edelman received the Nobel Prize in Physiology or Medicine in 1972 for work on antibody structure and subsequently devoted more than thirty-five years to understanding how the brain works. He has written several books on neural Darwinism and consciousness. A good place to start is *A Universe of Consciousness, How Matter Becomes Imagination* (2000). *Neural Darwinism: A Theory of Neuronal Group Selection* (1987) lays out his ideas in greater detail.

10. The notion of the Bayesian brain incorporates several threads whose origins date back at least twenty years. The individual who has contributed the most in recent years is Karl Friston at University College, London. I am not aware of good explanations in the popular literature, but three review articles that are a good place to start if one wishes to investigate further are: "The Bayesian Brain: The Role of Uncertainty in Neural Coding and Computation." David C. Knill and Alexandre Pouget, *TRENDS in Neuroscience* 27: 712–719 (2004); "The Free Energy Principle." Karl Friston, *Nature Reviews Neuroscience* 2: 127–138 (2010); and "Hierarchical Bayesian Inference in the Visual Cortex." Tai Sing Lee and David Mumford. *J. Opt. Soc. Am A.*, 20:1434–1448 (2003).

11. Karl Friston, "A Theory of Cortical Responses," *Philosophical Transactions of the Royal Society* 360:815–836 (2005), and "The Free Energy Principle" (2010), cited in note 10.

12. To explore Dr. Ed Sobey's work, go to [www.invention-center.com](http://www.invention-center.com).

## 11. Cultural Evolution

1. In *The Selfish Gene* (1976), Richard Dawkins discusses basic principles of population genetics and introduces the concept of the meme.

2. Eric Beinhocker, in *The Origin of Wealth* (2006), argues for the usefulness of the terms *amplification* and *deductive tinkering*. This book provides an excellent introduction to understanding economies as evolving complex adaptive systems.

3. Actually, the majority of practicing scientists are only passingly interested in the subject. Scientists almost never learn their trade by studying the philosophy of



science but rather by apprenticing with successful scientists. *The Logic of Scientific Discovery* (1968), Karl Popper, and *The Structure of Scientific Revolutions* (1962), Thomas Kuhn. Neither author was a practicing scientist; Popper was a philosopher and Kuhn a historian.

4. There exists an extensive literature on science as an evolutionary process. Some major contributors are Karl Popper, Donald Campbell, Stephen Toumlin, and David Hull. I recommend Popper's *The Logic of Scientific Discovery*, cited in note 3, and *Science as a Process, an Evolutionary Account of the Social and Conceptual Development of Science* (1988) by David Hull.

5. Eric Beinhocker, *The Origin of Wealth* (2006).

6. Ibid.

7. Dennett, *Darwin's Dangerous Idea* (1995).

8. Beinhocker, 2006.

9. David Sloan Wilson, *Darwin's Cathedral, Evolution, Religion, and the Nature of Society* (2002).

## 12. The Evolution of Complexity

1. Christoph Adami, Charles Ofria, and Travis C. Collier. "Evolution of Biological Complexity" in *Proceedings of the National Academy of Sciences* (2000). This technical article defines complexity to be the information theory equivalent of negentropy and shows that this measure increases in certain evolutionary systems. I object to this use of the term *complexity* because it is used to characterize how close a selected system is to an optimum, something quite different than the general meaning of complexity.

2. The distinction between structural and functional complexity used here was taken from "The Growth of Structural and Functional Complexity During Evolution" by Francis Heylighen in *The Evolution of Complexity* (1999). Heylighen has written widely on the subject of complexity and the various ways it might be defined.

3. Ibid.

4. The primary technical reference to depth is "Logical Depth and Physical Complexity" by Charles Bennett, which appeared in *The Universal Turing Machine: A Half Century Survey* Published by Oxford University Press in 1988.

5. Figure 12.1 is adapted from data presented in J. J. Sepkoski's "A Kinetic Model of Phanerozoic Taxonomic Diversity. III. Post-Paleozoic Families and Mass Extinctions" in *Paleobiology* (1984). This data has been widely cited, discussed, and analyzed. Similar data has been reported for terrestrial animals and vascular plants (P. W. Signor, *American Zoologist* [1994]). Although various corrections have been proposed, the basic conclusion that multicellular life has increased in diversity over long periods of time is clearly established by the fossil record.

6. Stephen J. Gould was very fond of pointing out that life on this planet is utterly dependent on bacteria and much less dependent on the existence of mammals. Variations on this theme can be found in many of his writings.

7. Steven J. Gould, *Full House: The Spread of Excellence from Plato to Darwin* (1996).

8. Daniel McShea, "Complexity and Evolution; What Everybody Knows" in *Biology and Philosophy* (1991). McShea has probably looked more extensively for some connection between evolution and biological complexity than any other contemporary biologist.

9. Figure 12.3 is adapted from one that first appeared in T. R. Gregory, "Macroevolution, Hierarchy Theory, and the C-value Enigma" in *Paleobiology* (2004). Current data can be found in the Animal Genome Size Database at <http://www.genomesize.com/index.php>.

10. Figure 12.4 is redrawn with permission from one first presented in "Morphological complexity increase in metazoans" by James W. Valentine, Allen G. Collins, and C. Porter Meyer, (1994) *Paleobiology* 20: pp. 131–142. More recent data suggest that the sharp bend seen at the earliest times is an artifact of incomplete data. Current estimates of the time of origin for the Porifera are now around 700 my before the present, and for the Cnidaria are now about 680 my before present. (Interested readers are recommended to: "The Cambrian Conundrum: Early Divergence and Later Ecology Success in the Early History of Animals" D. H. Erwin, et al., (2011), *Science* 334, pp. 1091–1097.)

11. Stuart Kauffman, *The Origins of Order* (1993) and the more easily understood *At Home in the Universe* (1995).

12. McShea, 1991.

13. Daniel Ashlock, personal communication. References to the five published examples are: Dan Ashlock and John E. Mayfield, "Acquisition of General Adaptive Features by Evolution" in *Evolutionary Programming VII*, V. W. Porto, N. Saravanan, D. Waagan, and A. E. Eiben eds. (1998); Daniel Ashlock, Elizabeth Blankenship, and Jonathan Gandrud, "A Note on General Adaptation in Populations of Painting Robots" in *Proceedings of the 2003 Congress on Evolutionary Computation* (2003); Daniel Ashlock and Brad Powers, "The Effect of Tag Recognition on Non-Local Adaptation" in *Proceedings of the 2004 Congress on Evolutionary Computation*, Vol. 2 (2004); David Doty, "Non-Local Evolutionary Adaptation in Gridplants" in *Proceedings of the 2004 Congress on Evolutionary Computation*, vol. 2 (2004); Daniel Ashlock and Adam Sherk, "Non-Local Adaptation of Artificial Predators and Prey" in *Proceedings of the 2005 Congress on Evolutionary Computation* (2005).

14. W. Ross Ashby, *An Introduction to Cybernetics* (1956).

15. See note 4.

16. Leigh Van Valen, "A New Evolutionary Law," *Evolutionary Theory* (1973). The Red Queen's hypothesis recognizes that organisms are in constant competition

with other evolving organisms and that the environment of every taxonomic group (species, genus, etc.) is constantly changing because of the evolution of its competitors. In such an environment, the only way to keep up is to continue evolving. The hypothesis was presented as an explanation for the observation that over a wide range of taxonomic groups, the probability of extinction is constant over time and independent of the time the group has been in existence. Extinction is assumed to occur whenever a taxon fails to evolve adaptations that allow it to match the changes that are occurring to its competitors.

17. Lathrop PhD Dissertation, *Computing and Evolving Variants of Computational Depth*, 1997, and also "Compression depth and genetic programs." In *Proceedings of the Second Annual Conference on Genetic Programming*, pp. 370–378. Morgan Kaufmann, San Francisco, 1997. The typical reader will not find these sources very understandable, but a quick look will give one an idea of the abstract nature of theoretical computer science.

18. Bennett (1988) likened logical depth to the "value" inherent in an object, and he tied value directly to the "amount of computational history evident in an object." Juedes et al. (1994) refer to "computational usefulness," Lutz (personal communication) to the "organization of redundancy," and Lathrop (1997), referring back to Bennett and to Juedes et al., variously saw depth to be manifest as "intricate structure," "computational work stored in the organization of  $x$ ," or to "organization embedded in a string by computation." Thus, we see that Bennett's use of value is based on the notion that information present in input can be organized by the process of computation into output structure that exhibits increased usefulness for further computation.

### 13. Past and Present

1. There are many books written by scientists about the origin of life. Probably the best place to start, even though it is rather old, is *Origins, A Skeptic's Guide to the Creation of Life on Earth* (1987) by Robert Shapiro. Two more recent sources are: *Life's Origin* (2002) edited by William Schopf, and *Life on a Young Planet, the First Three Billion Years of Evolution on Earth* (2003) by Andrew H. Knoll. A major breakthrough was announced in 2009, when John Southerland's group at the University of Manchester reported a plausible route for the chemical synthesis of pyrimidine ribonucleotides, required precursors to RNA (Powner, M., Gerland, B., and Sutherland, J., *Nature* [May 2009]). Prior to this, nucleotide synthesis was viewed as the greatest unknown in the prebiotic chemical origin of life. Conditions that produce amino acids, nucleotides, and sugars in the same broth have still not been demonstrated, but the pathway reported by Powner et al. is a significant step in that direction.

2. Sustained replication of a two-component RNA system was demonstrated in 2009. In this system two presynthesized RNAs with RNA ligase enzymatic activity are combined with four specifically designed RNA polymers that when joined produce more of the two ligases. When this system is fed the four precursor polymers, more and more ligases are spontaneously made. The system is seemingly capable of indefinite replication so long as it is provided with precursor molecules (T. Lincoln and G. Joyce, *Science* [February 27, 2009]).

3. Stuart Kauffman, *Reinventing the Sacred* (2008).

4. Bernd M. Rode, Daniel Fitz, and Thomas Jakschitz, "The First Steps of Chemical Evolution Towards the Origin of Life," in *Chemistry & Biodiversity* (2007). This review article provides a reasonably up-to-date review of various chemical origin theories but does not include the several breakthroughs reported in 2009 (see references in notes 1 and 2).

5. The idea is outlined in two of Stuart Kauffman's books: *Reinventing the Sacred* (2008) and *Origins of Order* (1993). A highly technical analysis is provided by E. Mossel and M. Steel in "Random Biochemical Networks: The Probability of Self Sustaining Autocatalysis" in *Journal of Theoretical Biology* (2004).

6. Harold J. Morowitz, Vijaysarathy Srinivasan, Shelley Copley, and Eric Smith, "The Simplest Enzyme Revisited, the Chicken and Egg Argument Solved" in *Complexity* (2005).

7. A mole, in chemistry, is defined as the amount of substance of a system that contains as many "elementary entities" (e.g., atoms, molecules, ions, electrons) as there are atoms in 12 g of carbon-12 ( $^{12}\text{C}$ ). A mole has  $6.0221415 \times 10^{23}$  atoms or molecules of the pure substance being measured. A mole will possess mass approximately equal to the substance's molecular/atomic weight in grams. Because of this, one can measure the number of moles in a pure substance by weighing it and comparing the result to its molecular/atomic weight. This explanation was copied from Wikipedia, but a similar definition can be found in any introductory chemistry text.

8. Matt Inlay, "Evolving Immunity, A Response to Chapter 6 of Darwin's Black Box". This internet article can be found at: [http://www.talkdesign.org/faqs/Evolving\\_Immunity.html](http://www.talkdesign.org/faqs/Evolving_Immunity.html).

## 14. The Future

1. Wikipedia gives an excellent summary of Moore's law with references and data tables.

2. Ray Kurzweil, *The Singularity Is Near* (2005). Many of the illustrations in the book are available on the book's website, [singularity.com](http://singularity.com).

3. A. M. Fennimore, T. D. Yuzvinsky, Wei-Qiang Han, M. S. Fuhrer, J. Cumings, and A. Zettl, "Rotational Actuators Based on Carbon Nanotubes," in

*Nature* (2003). A movie of the motor in operation can be found at: [http://www.livescience.com/technology/050412\\_smallest\\_motor.html](http://www.livescience.com/technology/050412_smallest_motor.html).

4. D. M. Eigler and E. K. Schweizer, "Positioning Single Atoms with a Scanning Tunneling Microscope," in *Nature* (1990). Other atom images can be found at the IBM STM image gallery: <http://www.almaden.ibm.com/vis/stm/atomo.html>.

5. The National Institutes of Health provides a basic introduction to cochlear implants at: <http://www.nidcd.nih.gov/health/hearing/coch.asp>. As these devices are commercially available, several vendors provide educational materials on the Web.

6. Actually this is a very odd sort of plot because as time passes, point number 4 moves up closer to the expectation line. For this point to fall on the line as drawn will require several hundred more years. The fundamental problem for making projections is that there is no way to know with certainty where exactly to draw a line when there are so few data points.

---

# Glossary

**Action potential** A characteristic pattern of ion flows and accompanying voltage changes that move along neurons (nerve cells). Action potentials are the primary way that neurons communicate with one another and with muscle cells.

**Adaptive immunity** Those aspects of the immune system that are activated after an infection becomes established in the body. Active immunity creates molecular defenses tailored to specific diseases that the body encounters.

**Adaptive landscape** A visual way of displaying the values of a large number of potential solutions to some problem in three dimensions. Adaptive landscapes when graphed often look like cartoon mountain ranges.

**Algorithm** A set of instructions for carrying out a procedure or solving a problem. In common usage an algorithm is a list of commands that cause a computer to do something specific.

**Algorithmic information** A measure of information that is particularly useful in computer science. A nontechnical definition of the algorithmic information content of a bit string is the length of the shortest possible binary (0s and 1s) input to a universal computer that will output the string.

**Algorithmically random** Incompressible, no amount of computation can reversibly reduce the size of an incompressible object.

**Alpha particle** An atomic particle consisting of two protons and two neutrons.

**Alternative splicing (RNA)** A form of RNA modification in cells is called splicing. During splicing an RNA molecule is cut in two places and the intervening material removed. When multiple splicing events occur on the same RNA, splice junctions can be combined in different ways resulting in different intervening regions of RNA being removed. What final RNA actually results from splicing depends on which beginning and end splice sites are brought together in the cell. The result is alternative RNAs being produced from a single gene.

**Amino acids** Chemical building blocks that make up protein molecules. Amino acids can be free or linked together in chains.

**Amino acid side chain** Amino acids have a central structure consisting of two carbon atoms and one nitrogen atom bonded together. The end carbon is part of an “acid” group, the middle carbon is also bonded to a hydrogen atom and to another chemical structure, and the nitrogen is part of an amino group. The chemical structure attached to the middle carbon is called a side chain.

**Amplification** A term that Eric Beinhocker uses to describe the propagation and multiplication of business plans and other information rich products of social evolution.

**AND** A Boolean function (see Boolean function) that takes a value of 1 (or true) when all inputs are 1 and otherwise takes a value of 0 (or false).

**ANN** See *artificial neural network*.

**Anthropocentrism** Viewing a situation from the perspective that human beings are most important, or viewing reality from a human-centered perspective.

**Antibody** A specialized protein produced by the adaptive immune system. Antibodies bind with high specificity to foreign molecular structures when they are present in the body. Antibodies are produced by B-cells and circulate in the blood and lymph. Certain antibodies are also secreted.

**Antigen** A molecule (usually a protein or polysaccharide) that the immune system responds to. Usually this response is due to the binding of an antibody or a T-cell receptor to the antigen.

**Artificial neural network (ANN)** A computational model in which simple processing units (nodes) are organized in layers and each node receives inputs from nodes in the next lower layer. The first layer receives input data, and every node sends outputs to nodes in the next higher layer. Node outputs are a summation or other simple transformation of all the inputs to that node. Typically ANNs have three layers.

**Atom** The smallest unit of matter associated with chemistry. Every element is characterized by atoms having a specific number of protons and electrons.

**Autocatalytic set** A term that Stuart Kauffman has used to describe a network of chemical reactions in which some of the chemicals present catalyze some of the chemical reactions. If the number of chemicals is large enough and a large enough fraction of them are catalysts for reactions in the network, then the

network can maintain itself indefinitely whenever it is provided with suitable chemical inputs.

**B-cell** The cell type in the body that produces antibodies.

**Back propagation** An analytical method used for training artificial (computer) neural networks. The method requires that the desired numerical output of the network be known and that the mathematical function that converts a network node's weighted input to its output (activation function) be differentiable (in calculus, a differentiable function is one whose derivative exists at every point in its domain).

**Bacteriophage** A virus that infects bacterial cells.

**Bayes optimal** A statistical result that is optimal from a Bayesian statistics perspective.

**Bayesian** Referring to a branch of statistics that originated with Thomas Bayes. This body of mathematical theory incorporates the idea that there exist prior probability distributions (probabilities that exist before any data is collected).

**Bénard cells** A regular pattern of circulation (convection cells) that forms when a thin layer of liquid is heated uniformly from below or above.

**Boolean** Referring to mathematical or logical concepts derived from the work of George Boole (1854). Boolean functions have two values: true or false (1 or 0).

**Cambrian period** The geological period lasting from 542 to 488 million years ago. Cambrian strata are the oldest that contain obviously animal fossils. In strata older than the Cambrian, animal fossils are small, rare, and difficult to associate with living animal phyla.

**Cambrian sea** Any sea in the Cambrian geological period.

**CAS** See *complex adaptive system*.

**Catalysis** A change in rate of a chemical reaction due to the participation of a substance called a catalyst. Catalysts are not consumed by the chemical reactions they catalyze. Catalysts generally allow reactions to proceed at lower temperatures than otherwise would be required. In biology enzymes are protein catalysts that are highly specific for particular reactions.

**Cell (in biology)** The basic organizational unit of all living organisms. Cells are always bounded by a biological membrane and depend on DNA to determine which proteins are present within them.

**Cellular automaton (CA)** A one-, two-, or multidimensional grid of "cells" (boxes with properties, and not to be confused with biological cells). A CA is updated periodically by a rule that determines the new state of each cell based on the current state of that cell and the states of nearby cells.

**Channel (membrane)** A protein embedded in a cell membrane that selectively allows small molecules or ions across the membrane.

**Chaos theory** A mathematical field that analyzes the behavior of a wide variety of dynamical systems characterized by high sensitivity to initial conditions.



**Chemical** Any phenomenon or condition in which atoms or molecules interact with one another.

**Chordates** Chordates are a large animal group that includes the vertebrates (animals with backbones), as well as the tunicates and lancelets. All chordates have the developmental structure known as a notochord, which in vertebrates is the embryonic precursor to the backbone.

**Circadian clock** A biochemical feedback loop that oscillates with approximately a twenty-four-hour period. Most cells contain at least one circadian clock.

**Code** A rule for converting one information-containing pattern into another. For example, the Morse code converts patterns of dots and dashes into letters, or letters into dots and dashes. The genetic code converts the order of nucleotides in DNA into the order of amino acids in proteins.

**Codon** A combination of three adjacent nucleotides in mRNA. See *messenger RNA*. There are sixty-four ways that four nucleotides can be arranged in threes (AAA, AAG, ATC, etc.). Each combination is called a codon. Sixty-one of these correspond to a particular amino acid in a protein, and three are “stop” codons.

**Coevolution** The situation that occurs when two or more species evolve together, each evolving in response to evolutionary changes in the other. Two classic examples are gazelles evolving to run faster to escape big cats and cheetahs evolving to run faster to catch gazelles; and flowers evolving red tube shaped flowers to attract hummingbirds and hummingbirds evolving long beaks to extract nectar from tube-shaped flowers.

**Complex adaptive system (CAS)** Any system in which parts (nodes) interact with other parts and have (some) memory of past interactions. In such systems future interactions depend in part on past interactions.

**Complexity** A loosely defined concept that quantifies the complicatedness of something. Complexity of something is generally thought of as some function of the number of parts of a system or object and the various interactions between those parts.

**Compression (in computation)** The computational conversion of one information-bearing object into another smaller one in such a way that the smaller can later be converted computationally (decompressed) back to the original larger form.

**Compression depth** A mathematical measure of the computational resources (time or memory space) required to reversibly compress a digital object.

**Computation** A process in which a pattern (usually of symbols) interfaces with a device in such a way that input-specific output is produced. Computations usually involve multiple steps.

**Computational chains** Linked computations where the output of one serves as input to the next.

- Computationally simple** Easy to compute from short input. Easy in this context means the computation requires few steps and little computer memory.
- Conservation of energy** The first law of thermodynamics: energy can neither be created nor destroyed (but it can change form). See *energy*.
- Convection cell** A coordinated motion observed in a liquid or gas when the bottom is heated or the top is cooled. The warm liquid (or gas) rises, and the cool liquid (or gas) falls in a regular pattern. A Bénard cell is a special type of convection cell.
- Convergent evolution** The evolution over time of what were initially very different organisms into those with very similar properties. A classic example is desert plants in different parts of the world that have evolved photosynthetic stems and whose leaves have become nonphotosynthetic spines.
- Coulomb's law** The force exerted by two point charges on each other is proportional to the magnitude of the two charges multiplied together and inversely proportional to the square of the distance between them.
- Countability** A countable set (in mathematics) is one that does not have more elements than there are natural numbers (1, 2, 3, etc.). An uncountable set is one that has more elements than can be matched one-to-one with a natural number. Any countable set has the property of countability.
- Critical state** Any state of a system where a small perturbation will cause a major transformation to occur.
- Cytoplasm** The portion of a cell that is inside the outer (plasma) membrane and outside the nucleus (if the cell has a nucleus).
- Depth** Used in this book to encompass several formal mathematical measures including “logical depth” (Bennett) and “computational depth” (Lutz). Informally, depth measures the difficulty of computing a digital object from simple input.
- Deoxynucleotide** A chemical unit of DNA. Deoxynucleotides are often informally referred to as nucleotides.
- Deterministic** Dictated totally by the situation, there is no choice and no uncertainty.
- Deterministic chaos** Generally what is described by chaos theory. The term is used to describe unpredictable behaviors that are completely described by deterministic equations. Typically such situations require infinite knowledge of the initial conditions if outcomes are to be predicted far into the future. See *chaos theory*.
- Deterministic computation** A computation that when repeated always produces the same output. A particular input always results in a particular output.
- Discrete** Not continuous; particulate; elements have separate values. In a system of discrete elements, the elements are distinct from one another in some way.
- DNA** Short for deoxyribonucleic acid, the molecular structure that genes are made of. Every living cell has (or once had) DNA that encodes information

necessary for cells to grow and divide. DNA molecules consist of long strings of smaller building blocks known as deoxynucleotides. Some DNA molecules are millions of deoxynucleotides long.

**DNA polymerase** An enzyme that synthesizes DNA molecules by chemically linking together deoxynucleotides using preexisting DNA as a template.

**DNA sequence** The specific order of the deoxynucleotides that make up a DNA molecule.

**Double helix** The typical molecular form of DNA in cells. A double helix consists of two single-stranded DNA molecules twisted around each other and held together by multiple hydrogen bonds. The two DNA strands have complementary nucleotide sequences so that when one has an “A” the other has a “T” and when one has a “G” the other has a “C” (and vice versa).

**Energy** Energy and entropy are two foundational properties of nature in classical thermodynamics. Energy is associated with mass and with potential for change. Energy changes are required whenever something new happens. The first law of thermodynamics states that energy can be neither created nor destroyed, but it can change form.

**Engine of complexity** A computational cycle that is present in all evolutionary (in a general Darwinian sense) processes. A diagram is given in figure 5.2. The cycle, when implemented, is capable of assembling complex instructions.

**Engine of construction** A cycle similar to the engine of complexity except that no copying occurs prior to selection. This cycle can accomplish impressive things, but is not as creative as the engine of complexity.

**Entropy** Entropy and energy are two foundational properties of nature in classical thermodynamics. For a given amount of energy, entropy measures a lack of capacity to do work. In the statistical mechanical interpretation, entropy measures the amount of randomness or disorder in a system. When a system is at equilibrium, the entropy of the system is the highest it can be and no work can be done by the system. Also, see *Shannon entropy*.

**Enzyme** A protein catalyst that is highly specific for a particular reaction. Enzymes also regulate the chemical reactions they catalyze. All natural enzymes are produced by living cells. See *catalysis*.

**Epistemology** The branch of philosophy that focuses on the nature and scope of knowledge.

**Equilibrium (in thermodynamics)** A situation in which no net change to a macroscopic system can occur without some input from outside the system. All forces and potentials are balanced at equilibrium.

**Equilibrium structure** A structure that is present in a system at equilibrium. Such structures do not change, but some exhibit repetitive motion such as a pendulum or a planet orbiting the sun.

**Eukaryote** Any organism that is based on the eukaryotic cell type. Eukaryotic cells are much more complicated than prokaryotic cells and contain numerous internal structures including a nucleus and mitochondria. All higher organisms (plants, animals, and fungi) are made of eukaryotic cells.

**EvoDevo** Evolutionary developmental biology. A branch of biology that attempts to deduce or reconstruct the evolutionary history of biological structures based on knowledge of the underlying molecular regulatory mechanisms.

**Evolution** This word has several meanings. The simplest is progressive change. This meaning is rarely used in this book; it is too vaguely defined. Another is Darwinian change. This is how heritable changes occur in lineages of organisms over time. A third meaning, and the most common usage in this book, is change that involves the engine of complexity. This is a generalized meaning of Darwinian change that is not restricted to living systems.

**Evolutionary** Based on evolution.

**Evolutionary algorithm** A category of computer algorithms that includes “genetic algorithms.” They always incorporate the engine of complexity.

**Evolutionary computation** A computation that makes use of the engine of complexity.

**Evolutionary epistemology** A school of philosophical study in which knowledge evolves by a process similar to natural selection. See *epistemology*.

**Evolved fit** Fit of one thing to another can come about in more than one way. When that way occurs by means of the engine of complexity, it is called evolved fit.

**First law of thermodynamics** Energy can neither be created nor destroyed, but it can change form. See *energy*.

**Gametophyte** A plant or plant tissue capable of producing sperm or egg cells. Plants exhibit a life cycle in which haploid (one copy of each chromosome) generations alternate with diploid (two copies of each chromosome) generations. Plants of a haploid generation are called gametophytes; plants of a diploid generation are called spermatophytes. Nearly all familiar plants are spermatophytes, and for seed plants, gametophytes are either pollen (male) or specialized tissues within the flower (female).

**Gene** A functional unit of DNA. Most often this term refers to a segment of DNA that encodes information necessary for the cell to form a specific protein.

**Gene expression** The production of a specific protein based on a particular gene sequence.

**Genetic algorithm** A computer algorithm that mimics biological evolution and that features “mutations” and “recombination.” These are a subclass of evolutionary algorithms.

**Genetic code** The rules that specify the correspondence of the order of nucleotides in DNA (and mRNA) to the order of amino acids in proteins. The

nucleotides comprising mRNA are read three at a time with each consecutive combination of three nucleotides (a codon) corresponding to a particular amino acid in a polypeptide. See also *codon* and *mRNA*.

**Genetic network** In cells the protein products of some genes regulate the production of proteins from other genes. The resulting network of regulations is called the genetic network (genes regulating other genes by means of the proteins they produce).

**Genetic recombination** A cellular process that involves the breaking and rejoining of DNA molecules in such a way that new DNA molecules are composed of segments of two or more older DNA molecules.

**Genome** The sum total of all the DNA sequences characterizing an organism. All organisms (unless they are an identical twin or a clone) have a unique genome.

**Genotype** A term used in biology for any aspect of the genetic makeup of an organism.

**Halting problem** In computer science it is sometimes impossible to know if an ongoing computation will eventually halt yielding output or will continue on forever without ever producing an output.

**Heisenberg uncertainty principle** A fundamental limit to the precision with which certain pairs of physical properties of an atomic-scale particle can be simultaneously known. An example is the more precisely one measures the position of a particle in space, the less precisely its momentum can be measured, and vice versa.

**Heritable variation (in a population)** In a natural population of organisms of the same kind, there always exist genetic differences. These differences across an entire population are called heritable variation. The genomes of each member of the population are subtly different. See *genome*.

**Hill function** A mathematical function whose solutions when plotted in three dimensions give the appearance of a very symmetrical hill.

**Hydrogen bond** An attraction between a hydrogen atom that is chemically (covalently) bound to one oxygen or one nitrogen atom in a molecule with another oxygen or nitrogen in the same or different molecule. Hydrogen bonds link otherwise distinct molecules (or parts of large molecules) together and are about one-tenth as strong as covalent bonds.

**Immunity** The state of an organism established by the immune system that provides sufficient molecular and cellular defense to protect the organism from infection by specific agents. Immunity is complex and includes innate immunity and adaptive immunity.

**Information** A simple general definition is “meaning conveyed by patterns of symbols.” More restricted mathematical definitions are provided by Shannon information and algorithmic information. See *Shannon information* and *algorithmic information*.

**Information body** A collection of symbols or features that is organized in some way so as to convey meaning to someone or something.

**Innate immunity** The bodily defenses against infection that are present before an organism is exposed to a specific disease-causing organism. Most potential infections never get started because of innate immunity.

**Instruction** Information organized in such a way as to cause something specific to happen or some structure to be built.

**Ion** An atom or molecule with a net (positive or negative) electrical charge.

**Ion channel** A protein embedded in a cell membrane that selectively allows specific ions across the membrane.

**Ion pump** A protein embedded in a cell membrane that translocates specific ions across the membrane. A pump requires a source of chemical energy to perform its task.

**Intentional stance** A term invented by Daniel Dennett to describe the situation where people use the language of human motivation to situations found in nature. An example is: "A fish has a tail in order to swim in water."

**IPCS** Iterated probabilistic computation with selection. A more formal way of describing the engine of complexity.

**Isolated system** In thermodynamics, a system is isolated if no energy (or mass) can enter or leave the system.

**Iteration** When a process repeats over and over with the result of each cycle serving as the starting point for the next.

**Kolmogorov complexity** Same as *algorithmic information*.

**Labile (chemical)** A labile chemical is one that breaks down readily; not chemically stable.

**Law (in science)** A simple, often mathematical, relationship. Laws are usually, but not always well established.

**Law of gravity (Newton's)** Masses always attract one another, and the force experienced by two (point) masses is proportional to the product of the masses and inversely proportional to the square of the distance between them.

**Learning** Data acquisition that involves generalization so that specific instances do not need to be remembered.

**Learning algorithms** Computer programs that allow a computer to learn something.

**Leukocyte** An immune cell in the blood or tissues; also called a white blood cell.

**Local minimum (or maximum)** In a landscape with hills and valleys, there is one depression that is the lowest and one hill that is the highest. A local minimum is a valley that is not as low as the lowest, but which is surrounded on all sides by higher terrain.

**Logic gate** A physical device that implements a Boolean function. See *Boolean function*.

**Macrophage** A subtype of leukocytes that move about the body like amoeba. They monitor all spaces ingesting debris and bacteria.

**Macrostate (thermodynamics)** Any state of a physical system at scales larger than atomic. The macrostate of a system is characterized by familiar properties such as pressure, temperature, and appearance.

**Major histocompatibility group (MHC)** A group of genes and the cell-surface proteins they code for that are the principal determinants of tissue type and transplant compatibility. MHC proteins are present on nearly all cells in the body and play an essential role in the recognition of foreign antigens by T-cells.

**Mass** A fundamental property of matter. Mass measures an object's resistance to acceleration and also determines how strongly an object is gravitationally attracted to another mass. Mass is closely related to weight but is not specific to the planet Earth.

**Matter** The "stuff" in our universe with rest mass and volume. Generally this means it is made of atoms. Matter has mass, which is related to energy by Einstein's famous equation:  $E = mc^2$  (where  $E$  is energy,  $m$  is mass, and  $c$  is the velocity of light in a vacuum).

**Meaning** Meaning is present whenever an object or action has a specific effect on something or someone else. Meaning implies one thing has significance to another.

**Meme** An idea, behavior, or style that spreads from person to person within a culture. The term was invented by Richard Dawkins in his book *The Selfish Gene* to make easier his discussion of cultural evolution. A meme is analogous to a gene in genetics, but does not have the same structure or precision of definition.

**Messenger RNA (also mRNA)** A form of RNA that carries a copy of a DNA sequence to ribosomes where the nucleotide sequence of the RNA is used to determine the sequence of amino acids comprising a polypeptide.

**Micelle** A structure that is formed by a broad class of molecules with structural similarities to detergents. Such molecules are water soluble at one end and water insoluble at the other. When placed in water they tend to form an elongated ball (a micelle) with the water soluble ends facing outward and the water insoluble ends facing inward, excluding water from the interior of the micelle.

**Microstate (in thermodynamics)** The instantaneous positions and momentums of all the atomic- or molecular-scale particles in a system.

**MHC** See *major histocompatibility group*.

**Molecular motion** Same as thermal motion. The spontaneous motions of molecules and parts of molecules that reflect the temperature.

**Molecular tiles** Artificial chemical structures, usually made of DNA, which when placed in solution carry out computations by spontaneously building physical structures.

**Molecular tiling** The process of molecular tiles fitting together to create structure.

**Molecule** A structure composed of two or more atoms chemically bonded together.

Molecules generally have properties very different from their atomic constituents.

**Momentum** The tendency of a moving mass to continue in the same direction at the same speed. The numerical value of momentum is the product of the mass times the velocity.

**Monomer** A monomer is a small chemical unit that serves as a building block of a larger molecular construct called a polymer. A polymer is made of many repeating monomer units. See *polymer*.

**Moore's law** The empirical rule in the electronics industry that the number of transistors in newly released microprocessors doubles every two years. The law has proven remarkably accurate for over forty years.

**Morphological** Relating to shape or form.

**mRNA** Same as *messenger RNA*.

**Mutation** Any spontaneous or random change in the DNA of an organism or in the code of a genetic algorithm.

**Mutation rate** How frequently mutations occur.

**Natural selection** In Darwinian evolution, nature determines which organisms of one generation live and reproduce to populate the next generation.

**Negentropy** The difference between the entropy of a system and the maximal (equilibrium) entropy of that same system.

**Network** An organized structure consisting of nodes connected to other nodes. Nodes can be almost anything.

**Neuron** Nerve cell. The cell type in the body that makes up nerves and ganglia and is responsible for long-range electrical communication within the body.

**Neuronal communication patterns** Networks of biological neurons communicating with one another.

**Neuronal group selection** A theory of brain formation and function proposed by Gerald Edelman. In this theory the gross anatomy of the brain is determined by genetics, but the details of neuronal wiring are determined by somatic selection during growth and development. During this process neuronal groups self-organize into functional modules. Synaptic selection continues after the primary formation of the brain and operates on functional modules as well as on individual neurons.

**NKS systems** NKS stands for a "new kind of science." A term invented by Stephan Wolfram to describe logical systems in which simple rules and inputs lead to apparent complexity.

**Node** The intersection points of a network. A node can be almost anything that is connected to or interacts with other nodes.

**Nonequilibrium structure** Structure that occurs in systems that are not at equilibrium. Such structures generally require a continuous flow of energy through the system. Without energy flow, the structure dissipates.



**Nonlocalized adaptation** A term invented by Dan Ashlock to describe a non-specific increase in competence observed in computer-based coevolutionary systems.

**Nucleotide** A chemical monomer unit of RNA (and informally of DNA). The monomer units of DNA are formally called deoxynucleotides. See *monomer*.

**Nucleotide chain** A chain of nucleotides in which the sugar group of one nucleotide is connected to the phosphate group of the next by means of a chemical (covalent) bond.

**One-max** A simple computer programming problem in which the programmer has a computer convert a string of 0s and 1s into a string of all 1s of the same length without explicitly telling the computer how to recognize a 1. Evolutionary algorithms do this easily if the computer is allowed to add.

**Order** Any deviation from randomness. Order indicates the presence of pattern or structure.

**Oxidation/reduction** A term for a chemical reaction in which electrons are transferred from one molecule to another. In an oxidation/reduction reaction, whenever one reactant (molecule participating in the reaction) is oxidized another is reduced. The reactant that receives an electron is reduced, and the one that donates an electron is oxidized.

**Phage** Shorthand for bacteriophage.

**Phenotype** A generalization used in biology to refer to any physical characteristic of an organism.

**Phospholipid** A member of a class of biochemicals that are major constituents of biological membranes.

**Phospholipid bilayer** A two-layered structure that is formed by phospholipids. The phospholipid bilayer is the organizing principle of biological membranes.

**Plasmid** A DNA molecule that is separate from and that replicates independently of chromosomal DNA. Plasmids can carry genes coding for various proteins.

**Point mutation** A spontaneous change that involves one bit or one nucleotide.

**Polymer** A polymer is a large unit, usually chemical, made of a large number of similar, or identical, “monomers” connected together. A pearl necklace is a nonchemical analogy.

**Polypeptide** In this book the term is used interchangeably with *protein*; a chain of amino acids.

**Polysaccharides** Large biological polymers (macromolecules) whose monomers are monosaccharides (sugars).

**Population genetics** The subdiscipline of genetics that studies how gene frequencies change over time in populations of interbreeding organisms. Population genetics describes mathematically how biological evolution works.

**Potassium channel** A protein structure that specifically allows potassium ions but not other ions to cross a biological membrane.

**Power law** A mathematical relationship in which the frequency of events is proportional to some power of the size of the event. When phenomena follow a power law, small events are much more common than large events.

**Precursor molecules** Molecules are always created in chemical reactions. In chemical reactions precursor molecules react to yield product molecules. The precursors are also called reactants. In a biological cell reactants are themselves produced in chemical reactions having other reactants. In this way long chains of reactions precede the production of most molecules. Any reactant in such a chain can be considered a precursor molecule to any later product in the chain.

**Probabilistic computation** A computation that includes at least one random step. Probabilistic computations yield different outcomes each time they are run even if each run employs the same input.

**Program (for a computer)** Instructions interpretable by a computer that when implemented cause the computer to carry out a specific task.

**Prokaryote** Any organism whose structure is based on a prokaryotic cell type. Prokaryotic cells are simpler than eukaryotic cells and do not contain a nucleus. Bacteria are the most commonly encountered prokaryotes.

**Promoter sequences** Segments of DNA that protein transcription factors bind to in cells. Transcription factor binding increases or decreases the rate at which RNA polymerase initiates the synthesis of RNA copies of an adjacent stretch of DNA. See *transcription factor*.

**Protein** Large biological polymer consisting of one or more chains of monomer units called amino acids. Proteins are responsible for most activities in cells including the building of structures and the catalysis of chemical reactions.

**Protein synthesis** The process by which proteins are assembled in a cell. The resulting order of amino acids in a protein is exactly determined by the order of nucleotides in a corresponding messenger RNA.

**Proteome** The network of proteins interacting with other proteins in cells.

**Pump (ion)** A protein embedded in a cell membrane that translocates specific ions across the membrane. A pump requires a source of chemical energy to perform its task.

**Purposeful information** A particular organization of information that has an intended result or outcome. Purposeful information is “for” something.

**Quantum mechanics** The mathematical theory that describes and explains physical phenomena at very small scales.

**Random number generator** Generally this term refers to a computer algorithm that outputs random numbers. If the computer is deterministic (as nearly all computers are) then the outputs are formally referred to as pseudorandom.

**Random sequence** A list of symbols that exhibits no predictable order.

**Random variation (in biology)** A term used in biology to indicate that mutations to DNA are made with no foresight as to what effect they might have on the organism.

**Randomness** Any situation where total knowledge of the current or previous states provides no information about what comes next.

**Recombination** A term used in biology and also in the context of evolutionary algorithms. In biology it refers to a molecular process in which two DNA molecules are broken and then recombined so that what was previously part of one molecule is now attached to the other. In computer science the term has a similar meaning except that some other sort of data structure substitutes for DNA. Recombination is also called "crossing over."

**Red Queen** In a coevolutionary system, each species must continuously evolve if it is to maintain its relative fitness among the other species in the system. The term is taken from the Red Queen's race in Lewis Carroll's *Through the Looking-Glass*. Says the queen running side by side with Alice, "Now, here, you see, it takes all the running you can do, to keep in the same place. If you want to get somewhere else, you must run at least twice as fast as that!"

**Reducing conditions (chemical)** Reducing conditions are situations dominated by reducing agents (see oxidation/reduction). In the biosphere the most common oxidizing agent is molecular oxygen, and the most common reducing agent is hydrogen that is not bound to oxygen. A reducing atmosphere is one that contains molecular hydrogen, ammonia (nitrogen bound to hydrogen), methane (carbon bound to hydrogen), and little molecular oxygen.

**Redundancy** Repetition in a system. In computer science a digital object with redundancy is one that can be compressed without loss of information.

**Reentry** A term coined by Gerald Edelman to describe "the ongoing dynamic interchange of signals that occurs in parallel between brain maps, and which continuously interrelates these maps to each other in time and space." Reentry depends for its operations on intricate networks of massively parallel reciprocal connections within and between neuronal groups. Edelman further describes reentry as a form of ongoing higher-order selection occurring in the brain.

**Resting potential** The normal unstimulated voltage that can be measured across a cell membrane.

**Ribosomes** Molecular structures occurring in all cells, they are the sites of protein synthesis in cells.

**RNA** Short for ribonucleic acid. A biological polymer of nucleotides that is structurally very similar to DNA but which has different functions in the cell. Like DNA, RNA can encode information.

**RNA polymerase** An enzyme that synthesizes RNA by linking together nucleotides.

**Second law of thermodynamics** The entropy of an isolated system must either increase or remain the same. An alternative statement is that heat always flows from hot to cold.

**Seed row** The first row of a cellular automaton. See *cellular automaton*.

**Self-assembly** Any process whereby a structure made of multiple parts comes together without external help. There is “no assembler.” Thermal motion usually provides the mechanism by which the parts move into position.

**Self-organized criticality** When a system naturally progresses to a state where it is poised between stability and instability.

**Sequences (of symbols)** A one-dimensional listing of symbols in which order matters.

**Shannon entropy** Shannon information.

**Shannon information** A mathematical definition of information introduced by Claude Shannon based on the probabilities of available possibilities; closely related to the definition of entropy in statistical mechanics.

**Shannon measure** Shannon information.

**Sierpinski gasket** A pattern consisting of triangles of all sizes generated by repeated application of the XOR Boolean function. See *XOR*.

**Simple input (to a computation)** Generally this means short input. Where input is not linear, it means modest in size.

**Slow growth** A property of depth. Deep objects can only be created from simple (short) input by means of extensive computation (i.e., slowly).

**Sodium channel** A protein structure that allows sodium ions across a biological membrane.

**Splicing (RNA)** Many RNA molecules in the cell are first synthesized as larger precursor RNA molecules. One of the several ways that these precursor molecules may be modified is by removing one or more internal segments (introns). The removal of introns and the rejoining of the RNA molecule is called splicing.

**Spontaneous folding (of a polypeptide)** A chain of amino acids spontaneously wraps back upon itself forming a compact structure. The exact three-dimensional path of the chain is determined by detailed interactions between amino acids and with the surrounding liquid. By this means every amino sequence has a characteristic compact shape.

**Stable states (of a network)** Many networks exhibit dynamic behaviors. When a dynamic network settles into an unchanging state, this is called a stable state. External inputs can often modify or disrupt stable states causing a transition to a new state.

**State (of a system)** One particular configuration of the components comprising a system.

**State probability** The likelihood that a particular state of a system will occur.

**Stochastic** Probabilistic, partially or fully random; possesses an element of randomness.

**Structure** The same as order and the opposite of randomness. In a structure there are fixed (or at least nonrandom) relationships between various parts.

**Structure for free** Formation of structure without following instructions.

**Supernetwork (in cells)** Cells are characterized by a network of biochemical reactions, a network of genetic regulations, and a network of proteins interacting with other proteins (the proteome). These three networks are aspects of a tripartite supernetwork that defines the nature of a biological cell.

**Sweet spot** When hitting a baseball, if the ball hits a particular part of the bat the ball goes farther and the bat does not sting the batters hands. This part of the bat is called the sweet spot. The term is used in other venues to mean a small region of a larger area where something works particularly well.

**System (in physics)** Anything physical that has a boundary.

**T<sub>4</sub>** A particular strain of bacteriophage.

**T-cell** An immune system cell type that produces T-cell receptors. Some T-cells detect and destroy virus-infected cells in the body; others modulate the actions of different immune system cells.

**T-cell receptor (TCR)** A protein that is produced on the surface of T-cells. T-cell receptors bind to foreign polypeptide (a chain of amino acids) antigens that are “presented” by MHC proteins on cell membranes. See *MHC*.

**Template** A physical structure that is used to form a complementary structure. One strand of the DNA double helix serves as a template for a new strand during DNA replication. The term is used with the same meaning as *seed row* when discussing cellular automata.

**Theory (in science)** An explanation of a phenomenon. Theories can be tentative or well established.

**Thermal motion (of molecules)** The spontaneous motion of very small structures that is the physical manifestation of temperature.

**Thermodynamics** That branch of physics that is concerned with energy and work.

**Transcription** RNA synthesis. In molecular genetics, transcription refers to the process of forming an RNA sequence that is a copy of a DNA sequence.

**Transcription factor** A protein that regulates synthesis of RNA from a DNA template. Generally this is accomplished by binding to DNA and interacting with RNA polymerase.

**Translation** In molecular genetics, translation refers to the process of protein synthesis.

**Transposase** An enzyme that is responsible for cutting out a small segment of DNA and inserting it at a new location on a larger DNA molecule.

**Transposon** A segment of DNA that is cut out and reinserted at a new location on a larger DNA by a transposase. Frequently the transposon carries with it the gene encoding the transposase needed for it to move. When that is the case the transposon has a high degree of independence and under appropriate conditions can move to a new cell or even to a different species.

**Tripartite supernetwork** See *supernetwork*.

**tRNA (transfer RNA)** A class of RNA molecules that carry amino acids to ribosomes and line them up according to the sequence of nucleotides in mRNA.

**Turing machine** A device, usually hypothetical, that was first used by Alan Turing in 1936 to prove certain principles of computer science. A Turing machine manipulates symbols on a strip of tape according to a few simple rules. It was the intellectual precursor to modern computers.

**Type I object** Something whose formation does not involve instruction use.

**Type II object** Something whose formation involves instruction use.

**Uncomputably many** A collection so large that computers, no matter how fast, cannot compute them all.

**Uncountably many** More than the number of natural numbers (1, 2, 3, etc.).

**Universal computer** A computer that can be programmed to simulate any other computer. A personal computer is a universal computer.

**Universal gas law** Pressure, volume, and temperature of a gas are related according to the relationship:  $PV = nRT$ . In this relationship,  $P$  is the pressure,  $V$  is the volume,  $T$  is the temperature,  $R$  is a constant that determines the units, and  $n$  is the amount of gas.

**Vesicle** In biology a closed ball-like structure made of biological membrane is called a vesicle. A vesicle encloses a trapped volume of water. Vesicles can also be formed from artificial phospholipid bilayers.

**Weights (of an ANN)** The nodes of an artificial neural network accept input from multiple sources, but these inputs do not necessarily carry the same weight. Some have more influence than others. A weight is a number that when multiplied times an input value determines how important it is. A zero weight means the input is completely ignored. See *artificial neural network*.

**XOR** Exclusive Or. A Boolean function that returns the value of 1 when one and only one of its inputs is 1 and otherwise returns the value 0. See *Boolean function*.

**ZIP file** A file that has been compressed using a ZIP compression algorithm. ZIP files are smaller than the original. The original file can be recovered by “unzipping” (a kind of computation) the ZIP file.



---

## References

- Adams, Douglas. 1979. *The Hitchhikers Guide to the Galaxy*. London: Pan Books.
- Adami, Christoph, Charles Ofria, and Travis C. Collier. 2000. "Evolution of Biological Complexity." *Proceedings of the National Academy of Sciences* 97: 4463–4468.
- Ashby, W. Ross. 1956. *An Introduction to Cybernetics*. London: Chapman & Hall.
- Ashlock, Daniel. 2006. *Evolutionary Computation for Modeling and Optimization*. New York: Springer.
- Ashlock, Daniel and John E. Mayfield. 1998. "Acquisition of General Adaptive Features by Evolution." In *Evolutionary Programming VII*. V. W. Porto, N. Saravanan, D. Waagan, and A. E. Eiben eds., 75–84. London: Springer.
- Ashlock, Daniel, Elizabeth Blankenship, and Jonathan Gandrud. 2003. "A Note on General Adaptation in Populations of Painting Robots." In *Proceedings of the 2003 Congress on Evolutionary Computation*, 46–53, IEEE Press.
- Ashlock, Daniel and Brad Powers. 2004. "The Effect of Tag Recognition on Non-Local Adaptation." In *Proceedings of the 2004 Congress on Evolutionary Computation*, vol 2. 2045–2051. IEEE Press.
- Ashlock, Daniel and Adam Sherk. 2005. "Non-Local Adaptation of Artificial Predators and Prey." *Proceedings of the 2005 Congress on Evolutionary Computation*. 41–48. IEEE Press.



- Bak, Per. 1996. *How Nature Works, The Science of Self-Organized Criticality*. New York: Copernicus.
- Bear, Mark F., Barry W. Connors, and Michael Paradiso. 2007. *Neuroscience*, 3rd ed. Baltimore: Lippincott, Williams and Wilkins.
- Behe, Michael. 1996. *Darwin's Black Box: The Biochemical Challenge to Evolution*. New York: Touchstone.
- Beinhocker, Eric. 2006. *The Origin of Wealth: Evolution, Complexity, and the Radical Remaking of Economics*. Cambridge: Harvard Business School Press.
- Bennett, Charles. 1988. "Logical Depth and Physical Complexity." In *The Universal Turing Machine: A Half Century Survey*. Rolf Herken, ed., 207–236. Wien: Springer Verlag.
- Borges, Jorge Luis 1962. "The Library of Babel," 51–58, In *Labyrinths*. New York: New Directions.
- Bray, Dennis. 2009. *Wetware, a Computer in Every Living Cell*. New Haven: Yale University Press.
- Campbell, Donald T. 1974. "Evolutionary Epistemology." In *The Philosophy of Karl Popper*, P. A. Schlipp, ed., Chapter 12, 412–463. Chicago: Open Court.
- Campbell, Neil A. and Jane B. Reese. 2008. *Biology*, 8th ed. San Francisco: Benjamin-Cummings.
- Carroll, Sean. 2005. *Endless Forms Most Beautiful, the New Science of Evo Devo and the Making of the Animal Kingdom*. New York: W. W. Norton.
- Compton, Arthur H. 1940. *The Human Meaning of Science*. Chapel Hill: University of North Carolina Press.
- Compton, Arthur H. 1935. *The Freedom of Man*. New Haven: Yale University Press.
- Cover, Thomas M. and Joy A. Thomas. 1991. *Elements of Information Theory*. New York: Wiley.
- Cyran, S. A., A. M. Buchsbaum, K. L. Reddy, M. C. Lin, N. R. Glossop, P. E. Hardin, M. W. Young, R. V. Storti, and J. Blau. 2003. "vrille, Pdp1, and dClock Form a Second Feedback Loop in the Drosophila Circadian Clock." *Cell* 112:329–341.
- Cziko, Gary. 1995. *Without Miracles*. Cambridge: MIT Press.
- Darwin, Charles. 1839. *The Voyage of the Beagle*. London: Colburn.
- Darwin, Charles. 1859. *On the Origin of Species by Means of Natural Selection*. London: John Murray.
- Darwin, Charles. 1871. *The Descent of Man, and Selection in Relation to Sex*. London: John Murray.
- Davies, Paul and Niels Henrik Gregersen, eds. 2010. *Information and the Nature of Reality*. Cambridge: Cambridge University Press.
- Dawkins, Richard. 1976. *The Selfish Gene*. Oxford: Oxford University Press.
- Dawkins, Richard. 1986. *The Blind Watchmaker*. New York: W. W. Norton,

- Dawkins, Richard. 1996. *Climbing Mount Improbable*. New York: W. W. Norton.
- Dawkins, Richard. 2004. *The Ancestors Tale: A Pilgrimage to the Dawn of Evolution*. Boston: Houghton Mifflin.
- Dennett, Daniel. 1995. *Darwin's Dangerous Idea: Evolution and the Meanings of Life*. New York: Simon & Shuster.
- Dennett, Daniel. 2003. *Freedom Evolves*. New York: Viking.
- Devaney, Robert L. 1990. *Chaos, Fractals, and Dynamics, Computer Experiments in Mathematics*. Boston: Addison-Wesley.
- Dorit, Robert. 1997. "A Review of *Darwin's Black Box: The Biochemical Challenge to Evolution*." In *American Scientist* 85; 474–475.
- Doty, David. 2004. "Non-Local Evolutionary Adaptation in Gridplants." *Proceedings of the 2004 Congress on Evolutionary Computation* vol 2. 1602–1609. IEEE Press.
- Dyson, George B. 1997. *Darwin Among the Machines: The Evolution of Global Intelligence*. New York: Perseus Books.
- Edelman, Gerald. 1987. *Neural Darwinism: A Theory of Neuronal Group Selection*. New York: Basic Books.
- Edelman, Gerald and Giulio Tononi. 2000. *A Universe of Consciousness: How Matter Becomes Imagination*. New York: Basic Books.
- Eigler, D. M. and E. K Schweizer. 1990. "Positioning Single Atoms with a Scanning Tunneling Microscope." *Nature* 344:524–526.
- Erwin, Douglas H., Marc Laflamme, Sarah M. Tweedt, Erik A. Sperling, Davide Pisani. 2011. "The Cambrian Conundrum: Early Divergence and Later Ecology Success in the Early History of Animals." *Science* 334:1091–1097.
- Fennimore, A. M., T. D. Yuzvinsky, Wei-Qiang Han, M. S. Fuhrer, J. Cumings, and A. Zettl. 2003. "Rotational Actuators Based on Carbon Nanotubes." *Nature* 424:408–410.
- Friston, Karl. 2005. "A Theory of Cortical Responses." *Philosophical Transactions of the Royal Society B: Biological Sciences* 360:815–836.
- Friston, Karl. 2010. "The Free Energy Principle." *Nature Reviews Neuroscience* 2:127–38.
- Gilbert, Scott F. 2006. *Developmental Biology*, 8th ed. Sunderland: Sinauer.
- Goldberg, David. 1989. *Genetic Algorithms in Search Optimization and Machine Learning*. Boston: Addison Wesley.
- Goldstein, Bernard, Michele Demak, Mary Northridge, and Daniel Wartenberg. 1992. "Risk to Groundlings of Death Due to Airplane Accidents: A Risk Communication Tool." *Risk Analysis*, 12: 339–341.
- Gordon, Peter. 2004. "Numerical Cognition Without Words: Evidence from Amazonia." *Science* 306:496–499.
- Gould, Steven J. 1989. *Wonderful Life: The Burgess Shale and the Nature of History*. New York: W. W. Norton.

- Gould, Steven J. 1996. *Full House: The Spread of Excellence from Plato to Darwin*. New York: Three Rivers.
- Greene, Brian. 2003. *The Elegant Universe: Superstrings, Hidden Dimensions, and the Quest for the Ultimate Theory*. New York: W. W. Norton.
- Gregory, T. R. 2004. "Macroevolution, Hierarchy Theory, and the C-Value Enigma." *Paleobiology* 30:179–202.
- Heylighen, Francis. 1999. "The Growth of Structural and Functional Complexity During Evolution." In *The Evolution of Complexity: The Violet Book of 'Einstein Meets Magritte' (Einstein Meets Magritte: An Interdisciplinary Reflection on Science, Nature, Art, Human Action and Society)*. Francis Heylighen, Johan Bollen, and Alexander Riegler eds., 17–44. Dordrecht: Kluwer.
- Holland, John. 1992. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*, 2nd ed. Cambridge: MIT Press.
- Holland, John. 1995. *Hidden Order: How Adaption Builds Complexity*. New York: Perseus Books.
- Holland, John. 1998. *Emergence: From Chaos to Order*. New York: Perseus Books.
- Hull, David. 1988. *Science as a Process: An Evolutionary Account of the Social and Conceptual Development of Science*. Chicago: University of Chicago Press.
- Inlay, Matt. 2002. "Evolving Immunity, A Response to Chapter 6 of Darwin's Black Box." Available at [http://www.talkdesign.org/faqs/Evolving\\_Immunity.html](http://www.talkdesign.org/faqs/Evolving_Immunity.html). Accessed January 15, 2013.
- Janeway, Charles, Mark Shlomchik, Paul Travers, and Mark Walport. 2004. *Immunobiology*. New York: Garland Science.
- Juedes, David W., James I. Lathrop, and Jack H. Lutz. 1994. "Computational Depth and Reducibility." *Theoretical Computer Science* 132(2):37–70.
- Kandel, Eric. 2007. *In Search of Memory: The Emergence of a New Science on Mind*. New York: W. W. Norton.
- Kauffman, Stuart. 1993. *Origins of Order: Self-Organization and Selection in Evolution*. Oxford: Oxford University Press.
- Kauffman, Stuart. 1995. *At Home in the Universe: The Search for Laws of Self-Organization and Complexity*. Oxford: Oxford University Press.
- Kauffman, Stuart. 2008. *Reinventing the Sacred: A New View of Science, Reason, and Religion*. New York: Basic Books.
- Knill, David C. and Alexandre Pouget. 2004. "The Bayesian Brain: The Role of Uncertainty in Neural Coding and Computation." *TRENDS in Neuroscience* 27:712–719.
- Knoll, Andrew H. 2003. *Life on a Young Planet, the First Three Billion Years of Evolution on Earth*. Princeton: Princeton University Press.
- Koschmieder, E. L. 1993. *Bénard Cells and Taylor Vortices*. Cambridge: Cambridge University Press.

- Kuhn, Thomas. 1962. *The Structure of Scientific Revolutions*. Chicago: University of Chicago Press.
- Kurzweil, Ray. 2005. *The Singularity Is Near*. London: Penguin Books.
- Lathrop, James. 1997. PhD Dissertation. *Computing and Evolving Variants of Computational Depth*. Ames: Iowa State University.
- Lathrop, James. 1997. "Compression Depth and Genetic Programs." In *Proceedings of the Second Annual Conference on Genetic Programming*, 370–378. San Francisco: Morgan Kaufmann.
- Lee, Tai Sing and David Mumford. 2003. "Hierarchical Bayesian Inference in the Visual Cortex." *Journal of the Optical Society of America A* 20:1434–1448.
- Leiman, P. G., S. Kanamaru, V. V. Mesyanzhinov, F. Arisaka, and M. G. Rossmann. 2003. "Structure and Morphogenesis of Bacteriophage T<sub>4</sub>." *Cellular and Molecular Life Sciences* 60:2356–2370.
- Lincoln, Tracy A. and Gerald F. Joyce. 2009. "Self-Sustained Replication of an RNA Enzyme." *Science* 323:1229–1232.
- Lloyd, Seth. 2006. *Programming the Universe: A Quantum Computer Scientist Takes on the Cosmos*. New York: Alfred A. Knopf.
- McShea, Daniel W. and Robert N. Brandon. 2010. *Biology's First Law: The Tendency for Diversity & Complexity to Increase in Evolutionary Systems*. Chicago: University of Chicago Press.
- McShea, Daniel W. 1991. "Complexity and Evolution; What Everybody Knows." *Biology and Philosophy* 6:303–324.
- Morowitz, Harold J., Vijaysarathy Srinivasan, Shelley Copley, and Eric Smith. 2005. "The Simplest Enzyme Revisited, the Chicken and Egg Argument Solved." *Complexity* 10:12–13.
- Mossel, Elchanan and Mike Steel. 2004. "Random Biochemical Networks: The Probability of Self Sustaining Autocatalysis." *Journal of Theoretical Biology* 233:327–336.
- Paley, William. 1802. *Natural Theology – or Evidences of the Existence and Attributes of the Deity Collected from the Appearances of Nature*. London: Richard Faulder.
- Pinker, Steven. 1997. *How the Mind Works*. New York: W. W. Norton.
- Popper, Karl "Of Clocks and Clouds." 1972. Chapter 6 in *Objective Knowledge: An Evolutionary Approach*. Oxford: Oxford University Press.
- Popper, Karl. 1968. *The Logic of Scientific Discovery*. New York: Harper.
- Powner M. W., B. Gerland, and J. D. Sutherland. 2009. "Synthesis of activated Pyrimidine Ribonucleotides in Prebiotically Plausible Conditions." *Nature* 459: 239–242.
- Purvis, Dale, George J. Augustine, David Fitzpatrick, William C. Hall, Anthony-Samuel La Mantia, and Leonard E White, eds. 2012. *Neuroscience*. 5th ed. Sunderland: Sinauer Associates.

- Rode, Bernd M., Daniel Fitz, and Thomas Jakschitz. 2007. "The First Steps of Chemical Evolution Towards the Origin of Life." *Chemistry & Biodiversity* 4: 2674–2701. Zurich: Verlag Helvetica Chemica Acta.
- Rothmund, Paul W. K., Nick Papadakis, and Eric Winfree. 2004. "Algorithmic Self-Assembly of DNA Sierpinski Triangles." *PLOS Biology* 2:2041–2052.
- Ruse, Michael. 2003. *Darwin and Design: Does Evolution Have a Purpose?* Cambridge: Harvard University Press.
- Schopf, William, ed. 2002. *Life's Origin*. Berkeley: University of California Press.
- Scientific American. 2002. *Understanding Artificial Intelligence*. New York: Scientific American.
- Sepkoski, J. J. 1984. "A Kinetic Model of Phanerozoic Taxonomic Diversity. III. Post-Paleozoic Families and Mass Extinctions." *Paleobiology* 10:246–267.
- Shapiro, Robert. 1987. *Origins, A Skeptic's Guide to the Creation of Life on Earth*. New York: Summit Books.
- Signor, P. W. 1994. "Biodiversity in Geological Time." *American Zoologist* 34: 23–32.
- Smith, Adam. 1776. *An Inquiry into the Nature and Causes of the Wealth of Nations*. London: William Strahan.
- Smolin, Lee. 1997. *The Life of the Cosmos*. Oxford: Oxford University Press.
- Valentine, James W., Allen G. Collins, and C. Porter Meyer. 1994. "Morphological Complexity Increase in Metazoans." *Paleobiology* 20:131–142.
- Van Valen, Leigh. 1973. "A New Evolutionary Law." *Evolutionary Theory* 1:1–30.
- Wilson, David Sloan. 2002. *Darwin's Cathedral, Evolution, Religion, and the Nature of Society*. Chicago: Chicago University Press.
- Wright, Sewall. 1932. "The Roles of Mutation, Inbreeding, Crossbreeding, and Selection in Evolution." In *Proceedings of the Sixth International Congress on Genetics*, 356–366.
- Wolfram, Stephen. 2002. *A New Kind of Science*. Champaign: Wolfram Media.
- Wolpert, Lewis, Jim Smith, Tom Jessell, Peter Lawrence, Elizabeth Robertson. 2006. *Principles of Development*, 3rd ed. Oxford: Oxford University Press.

---

# Index

- action potential, 110, 115, 231–34, 239–44, 250, 253; communication by means of, 244; flow, 232–34; transmission, 242–43
- adaptation, 30, 229, 285, 294, 302
- adaptive immune system, origin, 314–16
- adaptive landscape(s), 164–68, 293
- advancement: biological, 283–84; cultural, 257; equated to complexity, 26; evolutionary, 284, 286, 288, 289; scientific, 261, 342n7
- advancements, technological, 261, 268, 326, 331, 334
- adder, simple, 47–48
- agent: in complex adaptive systems, 229, 268; in evolutionary processes, 78, 299–301; as a finite state machine, 302; as replicator, 6
- age: of life on earth, 227, 306, 324, 336; of universe, 69, 133–35, 157
- agriculture: plant and animal breeding, 30, 203; role of selection, 202–3
- airplane: random modification, 129; odds of being hit and killed by, 134, 348n5
- albino, 124
- algorithm: computer, 41, 44, 193, 321–22; definition, 41, 44, 151; engine of complexity, 127–28, 169, 293; evolutionary (*see* evolutionary algorithm); genetic, (*see* genetic algorithms); as instruction, 41; implemented by tile sets, 61, 75; learning, 149, 169, 264; for making a screwdriver, 95–96; nonlocalized adaptation and, 294; one-max, 136, 153–58, 195, 292; paint bots, 294–96; part of a computer program, 126–27; as a representation, 44; role in computation, 45; sandpile, 86–89; Tartarus, 158–60; visualizing, 164–68

- algorithmic information, 40, 58, 67, 250, 280–282, 284, 301, 303; definition, 49–51; randomness, 51
- alternative splicing, 217
- altruism, 271
- amino acid: in antibody structure, 183–85; in protein structure, 99–105, 111, 194, 347n3; and origin of life, 308, 310–11, 354n1; side chain, 99
- amplification, in cultural evolution, 259, 265, 351n2
- analogy, role of in understanding, 253
- AND gate, 47–48
- animals: agricultural breeding, 30, 200–204; altruism, 271; appearance of major groups, 290; brains, 239–41; cell types, 217; culture, 256; deep structure, 300; development, 138, 175–76; evolution, 2, 191, 218, 284–86, 319, 336–37, 344n14, 352n5, 353n10; evolution of brain, 316–20; evolution of immune system, 314–16; fossils, 282–83, 286, 342n8; genomes, 197; genome size, 289, 353n9; heritable structures, 139; organs and appendages, 29; parasitic, 291–92; and plants; xi, 10; requirement for energy, 298; selection of, 129, 139; separate from humanity, 231; thought, 336
- ANN (artificial neural network), 146–49, 244–45, 318
- answers, pretty good, 166, 169, 293
- anthropocentrism, 284
- antibiotics, selection by, 201–2
- antibodies: fit, 36, 183; structure and production, 181–91
- antibody: affinity, 277; binding, 182–83, 186, 188; diversity, 185, 187; and evolution of immune system, 314–16; genes, 182–86, 188, 315; optimization, 292–93; structure, 183–84, 191; variable region, 184–85, 190
- antigen: binding, 182–84, 186, 188–89; definition, 182; presentation, 182, 188–89; self-antigens, 182, 185–86, 188–89
- Arabidopsis thaliana*, 139
- arbor, neuronal, 179
- Aristotle, 4, 12, 283
- artificial intelligence, 146, 348n2
- artificial neural network (*see* ANN)
- Ashby, W. Ross, 296–97
- Ashlock, Dan, 151, 158–60, 165, 294, 303, 339, 347n 348–49n3, 353n13
- assembly: random, 12, 135–36; virus, 105–10. *See also* self-assembly
- atmosphere out of chemical equilibrium, 10
- autocatalytic set, 310–13
- autocatalytic network, 314
- automobile: complexity example, 277–78; copying, 194; as a system, 17, 241
- avalanche, 83–88, 116, 268
- axon, 179–80, 190–98
- baby thought, 251
- back propagation, 148–49, 245
- bacteriophage, 105–10, 114, 139, 222, 226, 347n4
- bacteria, 106, 109, 141, 201, 217, 286; as virus host, 206–9
- bacterial: antibiotic resistance, 200–4; genetics, 200–2; population growth, 141
- Bak, Per, 84–85
- ball point pen parts, 278
- bartering, 265
- Bayesian brain, 245–49, 251, 253, 351n10
- Bayes optimal, 246
- B-cell, 181–91, 259; activation, 188
- behavior: atomic level, 89, 279, 334; of biochemical networks, 212–14; bulldozer behavior in Tartarus, 158–60; of cells, 218; of cellular supernetwork, 228; of complex systems, 207–9; of complex adaptive

- systems, 228–29; cooperative, 269, 271; economic, 268; of electronic circuits, 210–12; of evolutionary algorithms, 150; and functional complexity, 278; of genes over time, 258; human, 323, 342n10; improbable, 25; of nature, 7, 262; of neurons, 232, 244; of NKS systems, 51–58; of odometers, 86; power law, 84–85, 268; purposeful, 116; spontaneous, 86; as structure, 83–86; of tiling systems, 58–64; of Type II structures, 34
- Beinhocker, Eric, 264–69, 342n11, 344n12, 351n2
- Bénard cells, 79–82, 86, 89, 97, 117, 346n7
- big bang theory, 2, 69, 76, 133
- binary counting, 47–48
- binary object, 50, 279
- biochemistry first (origin of life), 312, 314
- biotechnology, 204–5, 326, 330–32, 334
- bit: amount of information, 157; in computation, 46–48; definition, 20, 343n4; information in nervous system, 111, 232–33, 242, 336; information in RNA, 313–14; length of a string, 20–21, 50, 136, 167, 281; in one-max example, 153–56; in physics, 8; predicting, 234; selected, 130, 143; smallest unit of information, 130, 200, 232, 258; space needed to store, 280
- blueprint: DNA as, 174; form of instruction, 13, 15, 91–93, 97; as a representation, 194; for a screwdriver, 93–94, 96; for a sculpture, 170, 172
- Boltzman, Ludwig, 18, 20
- bone marrow, 184–85, 188
- boolean networks, 208
- brain: architecture, 239, 245, 319, 329; Bayesian (*see* Bayesian brain); central activity, 243; communication is noisy, 236; as a computer, 26; computer interface, 334–36; engine of complexity, and, 251–54, 318–19; formation, 176–81, 189–91, 316–18; function, 239–41, 250–55; information encoding, 232, 241, 244, 251–52, 259, 320; information processing, 230, 232–34, 239–40, 248; internal connections, 177–78, 182; muscle coordination, 174, 232; origins, 316–20, 336; thinking up something new, 136; three ultimate sources of information in, 251; wiring, 176–81
- brain, human: circuits, 239; cannot create purposeful information out of nothing, 125; compared to computers, 146, 329, 334; complex adaptive system, 228; deep structure, 304; design strategy, 177; development, 190; example of a complex system, 207, 217; finding patterns, 243; imitation of, 329; information processing, 9; information encoding, 205, 321; number of neurons, 174, 176, 239; parallel structure, 233, 243, 245, 248, 253–54; probabilistic modification of brain patterns, 253; uniqueness, 177, 243
- brainstorming, 250
- bulldozer, in Tartarus, 158–60
- business plans, 265–69, 271–72
- CAD/CAM, 171
- Caenorhabditis elegans*, 317
- Cambrian explosion, 320, 342n8
- Campbell, Donald, 242, 345n18, 351n8
- catalysis, 308–11, 323
- catalytic: closure, 311, 314; polymers, 311
- causality, historical, 228
- chemical signaling, 232
- cell: bacterial (*see* bacteria); blast, 186; biology, 138–39; body, 180; cell-to-cell connections, 176–78, 234, 320; chromosomes in, 198; compared to a watch, 97–98; cytoplasm, 110; death, 176, 349n1; differentiation, 175, 218;



- division, 175, 186–89, 218, 226, 315–16; DNA codes for proteins, 105–18, 139, 142, 197; introducing new DNA, 137, 205, 332; DNA as measure of complexity, 288–90; requirement for DNA, 140, 225–28; DNA serves as instructions, 92, 97–105, 142–43, 193; egg, 124, 173–74, 195, 197, 199, 218, 228, 285; embryonic, 317, first common ancestor, 175–76, 227–28, 306, 311; growth, 139, 201, 215; immune system 181–91, 314–16; jellyfish, xiii; living computer, 25, 109; make up of larger organisms, 140, 170–91, 222–28; membrane, 110–14, 223–25, 316; migrations during development, 175, 218; muscle, 110, 232, 240; nerve (*see* neuron); as a network, 212–22, 227–28, 309, 311; number of types, 174, 217; number in human body, 174, 317; number in a nematode, 317; role of proteins in, 98–118; randomness in, 122; sensory, 239–40; seven features of living cells, 98; stem, 185, 188; subcellular parts, 97, 217; types, 105, 182, 218, 290–91; infected by viruses, 105–10; wall, 109
- cells, convection (*see* convection cells)
- cellular automaton (CA), 52–61, 66, 75
- cerebral cortex, parallel structure, 248
- chains: of computations, 125, 171–72; of events back to the beginning of time, 11, 118; of states, 45; of technologies, 328
- channels, potassium or sodium, 111–14
- chaos, 153; deterministic, 237–38, 350n4; edge of, 208
- chimpanzee typing, 137
- choices, yes/no, 24, 46
- chordates (origin of immune system), 315
- Christ, Jesus, 272
- Christian denominations, 270–71
- chromosome, 143, 197–200, 217, 221
- circadian clock, 221, 350n2
- clock, item of technology, 8, 172, 332–34
- clonal selection, 187, 189, 191, 315–16
- cochlear implant, 335, 356n5
- cockroach, 275, 286
- code, triplet, 102
- codon, 102–3, 347n3
- coevolution, 292–96, 301–4, 325–26
- coin tossing, 119–21, 134
- collective punishment, 271
- communication: within the body, 111, 217; in complex adaptive systems, 229; importance for culture, 256–60, 269; human, 30, 234, 256–60, 271, 320–21; and definition of information, 18–20; loops in society, 260; modes, 11; within nervous system, 97, 234, 238–41, 244, 250–55, 323; verbal, 259, 336
- communications industry, 18, 21
- competition: between cats and mice, 301; between potential solutions in the brain, 254; between religions, 270–72; and coevolution 292–96, 300; and depth, 301; Genome X-prize, 330–31
- complex adaptive system, 228–29, 268, 350n1, 351n2
- complex: design, xii; ideas, 256; structures, 4, 12, 70, 117, 173, 194–95; systems, 207–29; things, xiii–xiv, 1–5, 14, 32, 36, 51–52, 130, 137, 207, 280
- complexity: behavioral, 85–86, 207–8, 212; in biology, xiii, 8, 28–31, 80, 105, 138, 217, 275, 282–92, 241–42n4, 353n8, 353n10; in the body, 173–76, 217–18, 220; of brain patterns, 232–34, 239, 247, 251; of business plans, 269; costs of, 296–99, 301; cultural, 256; difficulty of defining, xi, 275–88, 343n7, 352n1, 352n2; of Earth, 10–13, 34; engine of (*see* engine of complexity); environmental, 292–99; ever increasing, 26, 274–76, 283,

- 288, 326; exceptional, 25; functional and structural, 177, 277–78, 352n2; generated by computation, 3–4, 24, 32–34, 51–58, 67, 121, 292–96; genetic, 217, 220; and information, 13, 37; irreducible, 344n13; Kolmogorov, 50, 280; long programs, 66–67; measure, 14, 22, 277–82; physical, 28, 32, 58, 68–90; purposeful, 34–38, 115–18, 341n1; relationship to randomness, 51; role of instructions, 13–15, 24–25, 35, 37, 67, 91–118, 130, 137, 172–73, 205; science, xii, 2, 247, 350n1; selection for, 276; and self-assembly, 105–9; type I and type II, 6, 15, 25, 34–35, 37
- compressibility, in a digital object, 51
- compression, 301–2; depth, 302–3
- Comte, Auguste, 7, 25
- Compton, Arthur, 237–38, 350n5, 351n6
- computational: perspective, 1–3, 26, 36, 38, 168, 242, 256, 279, 323; processes, hierarchy of (*see* structure hierarchy)
- computation: amount of, 280–81, 299–301, 354n18; by the brain, 239–41, 248, 329; cellular automaton, 52–58; challenges and responses, 299–301; complex outcomes, 4; compression, 301–2; conceptual glue, xi, 5; creative, 3; definition, 4, 9, 39–45, 67; deterministic, 49–51, 67, 121, 125; evolution as, xiv, 2–7, 23, 38, 116, 125–30, 135, 143–44, 192–206, 256–61, 272–74, 298, 305–24; evolutionary, 4, 6, 15, 25–26, 36–37, 276, 342n5, 348n3; evolutionary algorithms, 150–61, evolutionary programs, 161–54; general concept, 39–45; halting problem, 282; inseparability from physical process, 59, 64, 66–67, 86; iterated, 32–33, 125–28; living, 25, 137–44, 324, 344n10; long division, 43; long programs, 66, 92; mechanism, 45–48; by neurons, 239; one-max, 153–58, 292; physical process as, 32; of physical structure, 58–65, 75, 90, 107; principles of, 51–52, 67; probabilistic, 23–24, 46, 116, 125–29, 324; relationships created in, 280; reversible, 46; role of representation, 41; sandpile, 86–88; simulation of nature, 34; steps of, 52; Tartarus, 158–60; technology, 326–29; theory, xii, xiv, 32, 235; three parts of, 40; usefulness, 304, 354n18; visualization, 164–67
- computer: advantages over the brain, 146; cell as, 109, 344n10; code, 25, 293, 295, 302, 321, 323; copying by, 242; creation of, 37, 335; desktop, 2, 5, 7, 50, 66, 136, 242, 329; deterministic, 23, 25, 45, 46; digital, 16, 37, 44, 46, 48–49, 62, 177, 244, 279, 329, 335; drawing, 33; earliest, 9, 44; experiments, 200, 294, 302; file, 21, 302; language, 44, 49, 50, 322; learning (*see* learning); models, 244, 247, 291; output, 41, 70; power, 326–29; programs, 28, 33–34, 41, 86, 126, 143–44, 153, 161, 171, 192, 281–82, 294, 344n11; scientists, 49, 58, 91, 145, 150, 152, 204, 279, 293, 326; simulations, 311, 336; states, 42, 46, 67, 280; theory, 7, 32, 40, 66, 120, 234, 279–81, 354n17; universal, 48, 50, 54; universe as, 58
- computer science, field of, xiv, 2–4, 7, 9, 16, 23, 31–34, 39–42
- computing technologies, five, 326–27
- concepts, recirculation in society, 273
- configurations, role of alternatives in information theory, 18–22, 45, 193, 241; of artificial neural networks, 146–47; equilibrium, 73–74, 110; initial, 67, 70, 88, 93, 95, 159; interacting with other configurations, 64, 67, 241;

- internal relationships, 121–22; low probability, 93; of parts, 12, 70, 241; of a system, 18, 32, 45, 67, 241
- connections, in the brain, 146, 177–82, 190–93, 233, 239, 242–43, 247, 252, 316–17
- connectivity patterns, in the brain, 97, 233–34, 242–45, 319, 321
- consciousness, 231, 349n2, 351n9
- conservation: of energy, 27, 75–77, 82; of momentum, 75–77, 95
- constraint, on outcomes, 137
- control: in engineering, 296–97, 300; by enzymes, 215, 308; DNA sequences (*see* promoter); of the engine of complexity, 192–206; gene, 217–19; of muscles, 232, 240
- controllers, 158–59, 161, 295–96
- convection cells, 79–80
- cooperation, 229, 266, 271; among cells, 228
- cooperative: advantage, 269, 271; behavior, 269, 271; community, 271
- copying: computer, 127; DNA, 195, 219, 225–26; errors, 24, 126, 181, 191, 193, 195, 259; in evolutionary systems, 23–24, 176, 193, 196, 259–60, 272–74, 321–22; iterated, 126, 172; and learning, 146; repetitive, 171–73; role in culture, 259; with selection, 24, 176, 181, 189–191; three-dimensional objects, 191, 193, 194; three ways, 194
- corn (maize): genetically engineered, 204; number of chromosomes, 198; yields, 203
- corporations, evolution of, 265–69
- costs: of complexity, 297–98, 301; decreasing due to technology, 326–31, 335
- Coulomb's law, 72–75, 346n3
- countability, 236
- counting, 130–31, 236; up to two, 130, 348n2
- c-value, 288–89, 353n9
- creativity: and the engine of complexity, xiii, 130, 190, 231, 255, 325; of evolutionary processes, 3–4, 6; human, (*see* human creativity); of life, 3, 255; role of randomness, 7, 235–36, 238, 255, 315, 325; source of, 7
- critical states of a system, 84–85, 88–89
- crossover (*see* recombination)
- cultural change, 257–61
- culture: human, 37, 193, 256–61, 276, 347n6; western, 275, 283
- cyanobacteria, 286
- cytoplasm, 109–10, 224
- Cziko, Gary, 36
- crystal, salt, 71–75
- dairy cow milk production, 203
- Darwin, Charles, 28, 137–38, 200, 257–58, 307–8, 345n17, 348n6
- Darwinian: process, xiii, 4, 144, 341n4, 351n9; theory of evolution, 143, 283–84
- Darwinism, 6
- Dawkins, Richard, xii, 4–7, 24, 35, 98, 227, 258, 341n3, 342n11, 344n9, 344n13, 345n17, 347n2, 347n6, 351n1
- data: and the brain, 240, 242, 245–48, 251–52, 255, 319; extra, xii; input to a computation, 32, 41, 45, 48, 52, 54, 58, 126–27, 146–47, 280, 294, 298, 302
- deductive tinkering, 264, 266, 268, 351n2
- depth: aspect of complexity, 26, 280–82, 284, 299–304, 326, 352n4, 354n17; of biological structures, 299–302, 304; of challenges and responses, 299–300; definition, 280
- deities, 270
- deletion, genetic, 188, 197, 199, 255
- dendrites, 179–180, 190–91
- Dennett, Daniel, 6–7, 35, 44, 268, 342n6, 344n12, 345n2, 345n16, 348n3

- development, biological, 25, 29, 38, 138–39, 175–81, 189–91, 195, 315, 317, 319, 322, 349n1
- determinism, 8, 255, 318
- differentiation (*see* cell differentiation)
- digital revolution, 2, 257, 269
- discreteness, 236
- disorder, 119–22, 126, 129, 137, 277, 298
- DNA: amount in cells, 288–89; aspect of cells, 98, 104, 139–40, 174, 176, 185–90, 205, 227–28, 289; base pairing to RNA, 104; base pairing rules, 102, 225; in chromosomes, 197–99; encoding information, xiii, 3, 8, 23, 25, 38, 99, 102, 111, 114, 128, 142–43, 175, 177, 191, 193, 200, 215, 251, 259, 308; encoding instructions, 13, 92, 97–99, 104–5, 109, 174–75, 220, 222, 227, 318; human, 174–75, 177, 195, 218, 289, 291, 317; hyper mutation, 186; molecular structure, 100, 102, 124; molecular tiles, 58–62; as part of the genetic regulatory network, 215–22, 228; polymerase, 225–26; random changes, 38, 196–99; rearrangement in immune cells, 184–85, 188; recombination, 197–99; regulatory, 291; repair of, 102; replication, 194, 225–26; repository of the rules of development, 174, 322; sequencing, 330–31; structure, 102, 194, 225–26; synthesis, 137, 205; virus, 106–10
- DNA-codes-for-proteins-that-make-organisms-what-they-are paradigm, 97–114, 139–40, 179, 185, 215, 228
- DNA sequences, analysis of, 29, 315; depth of, 303; relationships to other DNA sequences, 30; selection of variants, 142, 195, 199
- DNA tiling (*see* molecular tiles) dogmas, religious, 31, 259, 270, 272, 283
- do-gooder, 271
- double helix (*see* DNA: structure)
- duplication, genetic, 98, 139, 197, 199
- early Earth models, 306–8
- E. coli*, number of genes, 215
- economic: benefits, 270–71; growth, 261; networks, 269; theory, 265, 269
- economies, 6, 265–70, 276, 351n2
- economy modern, 271, 274
- Edelman, Gerald, 177, 243, 344n12, 349n2, 351n9
- edge of chaos, 208
- edge: of a graph, 208–9, 214–16, 220, 228–29, 267; logical, 215; regulatory, 215–16, 220, 229
- Edison, Thomas, 249
- electronic circuits, 209–12
- emergence, 229, 268, 350n3, engine of complexity: basic description, 125–30; and the brain, 250–55; coevolution, 292–96; in computers, 145–69; control of, 192–206; creating improbable things, 135–36, 143, 229; and creativity (*see* creativity); and drive for continuation, 117; future, 336–38; generation of complexity, 275, 292; in the immune system, 181–89; importance of mutation rate, 157, 196; information assembly, 22–26, 136, 171; in life, 137–44; optimization, 292–96; origins, 305–23; and purpose (*see* purpose); in society, 256–74; three basic elements, 193; three requirements, 259; trick, 150, 169; why small changes are better than large ones, 24
- engine of construction, 170–76
- energy: of activation, 213; chemical, 11, 75, 98, 110–11, 117; conservation of, 27–28, 75–77, 82, 95; flow through a system, 71, 89, 116–17; kinetic, 75–76, 78; levels, 73–74, 346n3; potential, 75–76, 84, 110, 116–17; from sunlight, 10, 71, 306;

- thermodynamic, 17–18, 71, 246, 297–98
- environment: of the cellular network, 228; as challenge, 298; in coevolution, 292, 302; cultural, 326; depth, 300; early Earth, 308; economic, 268–69; of engine of complexity, 128–29, 272, 276, 292, 324; entities, part of, 300–2, 326, 354n; external to cells, 174–76, 178–80, 183, 185, 191, 222–23; fit to, 35, 145; human, 11, 15, 236; information about, 145, 149, 343n3, 343n7; internal to body, 296; internal cellular, 98, 111, 195, 215, 220, 224–25, 227; law of requisite variety, 296–97; of organisms, 30, 139, 141–43, 196, 243–44, 276, 283–85, 316, 318–19; of parasite, 292; of thunderstorms, 80; unchanging, 292; unpredictable, 297; of viruses, 110, 139
- entities, in an evolutionary system, 23–24, 116, 144, 127–30, 276, 292–302
- entropy: Shannon, 20–22, 343n4; thermodynamic, 17–18, 22, 277, 297–98, 343n7
- enzyme: in biochemical networks, 216, 221, 228; and catalysis, 215, 221, 308–9; DNA polymerase, 225–26; protein, 99–102; RNA ligase, 314; RNA polymerase; virus, 106
- epistemology, evolutionary, 241–44, 250, 351n8
- equilibrium: chemical, 10, 71; far from, 71, 80, 115; non-equilibrium phenomena, 78–83; thermodynamic, 17, 71–75, 78, 89, 110, 277
- error: in biological systems, 143; during copying, 24; correction, 62; in DNA replication, 196; in DNA sequencing, 330; hypermutation, 186; role in probabilistic computation, 129; in protein synthesis, 103; in tiling systems, 62; trial and, 249, 319
- EvoDevo (evolutionary developmental biology), 29, 139, 348n7
- evolutionary algorithms, 6, 150–69; climbing the hill function, 164–66; coevolution and, 293; complexity and, 275–76; optimization and, 292; origin of, 321, 323–24, 336–37; search speed up, 156; selection in, 200–4; sensitivity to mutation rate, 157; six elements, 150; to solve one-max, 154; structure of, 151; visualization of solution space, 167–68
- evolutionary computations, xiv, 2–4, 6, 15, 25–26, 342n5; creating instructions, 25, 36
- evolutionary history of animal brains, 235, 318
- evolutionary processes, five elements, 23–24
- evolutionary systems, 7, 22–26; coevolution, 292–96, 300–4, 326; five systems, 323; and learning, 16; nested, 322; operating on the “edge of chaos,” 208–29; planets with, 11; predictably increase complexity, 326; time measured in generations, 37
- evolution: in the brain, 244–55; as computation, xiv, 125–30, 143; in computers, 150–69; convergent, 326; and creativity, xiv, 4, 6–7, 130, 255; cultural, 256–74, 320–21; definition of, 4, 127–30, 143; diagram of, 126, 143, 253, 260; generalized, xiv, 1–2, 6, 36, 143–44, 167, 256; and information, 2–5, 26, 145, 153; law of, 5–6, 28, 341n4; of life, xiii, 2, 6, 36, 137–44, 197, 283, 292–93, 305–14; of life, seven principles, 140; major innovations and, 138; mechanism of (*see* engine of complexity); of microorganisms, 200–2; theory of, xiv, 29–30, 138, 143–4; thermodynamics of, 5; within the body, 181–91

- experiment: computer, 200, 294–95, 302–3, 350n4; in science, 26, 30–31, 138, 262–4
- extinction, 84, 129, 196, 265, 269–71, 291, 352n5, 353n16
- feedback loop: in the brain, 242; positive, 23, 233, 301, 310
- finite state machine, 302
- fish, 29, 35–38, 145, 174, 282, 284–85, 294
- fit: evolved, 16, 36, 38, 145–46, 263, 266, 345n18; to something, 1, 12, 35–36, 91, 97, 115; order, 269
- flower color, 139
- formula: chemical, 213–14, 216; mathematical, 162
- fossil record, 227, 284–86, 288, 290, 306, 352n5
- future, non-random change restricts, 196
- free energy, 11, 246
- free-energy principle, Bayesian brain, 247–52, 351n11
- free will, 255, 350n5
- Friston, Karl, 246, 351n11
- future potential, restriction of, 196
- Galileo, 31, 236
- gas: cloud, 68, 75–77; greenhouse, 306; law, 81; symmetry of, 279
- gazelle, 298–99
- gene: antibody, 182–86, 315; as an instruction, 13; as an entity, 23; as a hereditary unit, 102, 106, 111, 117, 124, 138, 140–41, 175, 177, 197–204, 227–28, 258; number, 177, 217–18, 290–91; protein coding DNA sequence, 97–105, 194, 202, 216–19, 221, 258, 289; regulatory, 29, 218–21; T-cell receptor, 188–89, 315; therapy, 332
- generalization (*see* learning)
- general theory of evolution, xiv. *See also* engine of complexity
- general relativity, theory of, xiii, 8, 263
- genetic: algorithms, 152–60, 167, 204, 298, 302, 349n2; code, 227; engineering, 200, 204–5, 331–32; networks, 216–22, 227, 290–91; programs (subclass of evolutionary algorithms), 161–64; regulation, 216–21, 227–28, 291
- genetics: agricultural, 203–4; bacterial, 200–2; molecular, xiii, 97–110, 138, 347n3; population, 25, 140–42, 197–99, 258, 270, 351n1
- genome, 197, 205; human, 218, 220, 289, 330–31
- geometry, 74, 117, 191, 193, 213, 236, 239
- Gibbs, Willard, 18
- God, 35–36, 115, 137, 270–71, 283
- good tricks, 268–69, 271
- Gould, Stephen, J., 286, 342n8, 353n6, 353n7
- graph, 85, 146, 164–65, 208–14, 286–91, 332, 337
- great principles of computing, 16, 343n2
- halting problem, 282
- Heisenberg uncertainty principle, 238
- heritable change in biological systems, major sources, 199
- heritable variation, 140, 142–43, 196
- Heylighen, Francis, 278, 352n2
- hill function, 164–67
- Hitchhiker's guide to the galaxy*, 122
- Holland, John, 150, 348n2, 350n3
- hominin, 286
- horizontal gene transfer, 199
- horseshoe crab, 275, 286
- human being, complexity of, xi, 13, 98, 173, 217, 277, 282–84, 317
- human body: copying, 195; formation of, 170–76; formation of, four things that happen, 218; number of cell types, 217; number of polypeptides, 217; role in instruction use, 97

- human brain (*see* brain, human)
- human computer interaction, educational programs, 334
- human: creativity, 255; culture, 37, 193, 256–74; genome project, 289, 330; formation of hand, 174–76; generation of random numbers, 234–36; language, 322–33; learning, xiii, 205, 230–55, 318–21; memory is fallible, 273; mind, 230–31; organizations, 269, 279; population growth, 30, 141; social institutions, 9, 11, 206, 270, 273, 276, 320, 342n11; society, 8, 135, 272–74; technology, 2, 66–67, 192–95, 255, 325–37; thought, 230–55, 264, 318–20, 324; thought simulation, 335–36
- human DNA: number of genes, 177, 317; number of nucleotides, 177, 330; regulatory, 218, 291
- humans, number of, 131, 133
- hydrogen bond, 225
- hypermutation, 186–87, 189, 191
- hypotheses: Bayesian brain, 244–48, 251; multiple in the brain, 248; Popper, 231; in science, 250, 261–64, 350n2
- IBM in atoms, 333, 356n4
- ideas, generation of new, 236, 250, 264
- immune system, 181–89, 292–93, 314–16
- immunity: adaptive, 181–89, 193, 305, 314–16, 336; cellular, 181, 188–89; humoral, 181–87; innate, 181, 315
- improbability (*see* probability)
- infection, 2, 105–10, 181–89, 201–2, 276–77, 292–93
- infinite improbability drive, 122
- information: accumulation, 3, 26, 127–30, 140, 143, 145, 152, 254; action potentials and, 111, 232–33; age, xiii, 8; algorithmic, 40, 50–51, 58, 67, 250, 280, 282, 284, 301, 303; bodies of information, 22, 26, 37, 116, 136, 147, 150–53, 171, 193, 204, 206, 250, 264, 272; bodies, wildly improbable, 250; in brain, 9, 205, 230–34, 241, 251–52, 320–21, 342n10; in computation, 39–42; conservation during computation, 40; content of thoughts, 231, 236, 242, 250, 255; copying, 24, 146, 242, 259, 273, 319–21, 336; cultural, 256–61; definition, 16, 18–22, 49–51; encoded in instructions, 6, 13, 15, 34, 36, 116, 205, 312, 322; encoded in DNA (molecular biology), 99–105; encoded in RNA, 309–14; and evolution, 5, 9, 22; extra or special, xiii, 13, 15, 66, 92, 96–97, 116; first (origin of life), 312–14; flows, 214–15, 223; manipulation, 3–4, 9, 39–40, 45–49, 171, 195, 324; new, 130, 154, 159, 167; organized, xii, 6–7, 13, 116, 125, 304, 354n18; in physical laws, 72; inherent in physical structure, 171–72, 195; purposeful, 15, 23, 37–38, 92, 97, 107, 125, 136, 255, 311, 313; random generation, 157; required to make screwdriver, 96–97; science, xiii, 8, 20, 22, 41, 232, 251, 264, 343n; Shannon, 21–22, 40, 50, 58, 67, 250, 277, 279–80, 284, 303, 343n4, 343n7; source of new, 149, 169; theory, 26, 178, 284, 352n1; three ways to define and measure, 67
- informational perspective, 4, 8, 23, 45, 111, 172, 251, 264, 270
- information assembly by the engine of complexity (*see* engine of complexity)
- initial conditions, unexceptional, 38, 67, 81, 83, 89, 115
- input: deep, 299; description of output, 121, 280–81; to the engine of complexity, 127–29; long, 66–7; to physical structure formation, 32, 75, 81; random, 23, 255; relationships between inputs and outputs, 34; sensory, 231, 240, 245–46, 318, 323;

- shorter than output, 121, 281;  
shortest, 49–51, 280; simple, 52–58,  
66–67
- instructions: assembly, 24; as business  
plans, 265–69; as complicated rules,  
6, 34, 115, 172; definition, 13;  
DNA sequence as, 92, 98–110, 174,  
227, 308, 318; examples, 91–118;  
have a purpose, 35–36, 38, 115–16;  
importance of, 13; improbability of,  
130–37; long, 37, 66–67, 92, 130,  
135–37; magic of, 25, 116, 125,  
192, 195; origin of, 6, 23, 125–30;  
representation of a method, 194–95;  
for a screwdriver, 95; and type II  
objects, 15, 25, 34, 36–37; wildly  
improbable, 115, 130–37
- intentional stance, 35, 345n16
- internal relationships: in objects, 122,  
280, 299; within a sequence, 121
- invention, 249–50; role in technology, 328
- ion, 73–74, 110, 114; channels 111–14;  
pumps, 111–14
- iterated prisoner's dilemma, 302
- iteration: in computation, 125–26,  
154–56; IPCS, 127–8; in structure  
hierarchy, 172–73
- “it from bit”, 8
- jellyfish, xii, xiii, 290, 320
- jigsaw puzzle, self-assembling, 64–65, 67,  
107
- Jones, Jim, 270
- Johnson, Nate, 92–94, 347n1
- Kauffman, Stuart, 208–11, 346n3,  
350n1
- Kuhn, Thomas, 261, 342n7
- knowledge: acquired, 16; encoded in  
brain, 230–50; fit with nature, 263;  
prior, 137; scientific, 261–65
- Kolmogorov, Andrei, 50
- Kolmogorov complexity (*see* complexity)
- Kurzweil, Ray, 326–36, 355n2
- language: computer, 44, 50, 322; of DNA,  
99, 103, 109, 179; evolution of, 276;  
importance for human culture, 256–57,  
260–61, 321, 323; and information,  
250; mathematical, 31; representations  
in different languages, 44
- Lathrop, James, 302–3, 354n18
- law: Coulomb's, 72–76, 346n4; of  
evolution, 5–6, 341n4; of gravity,  
7, 26–27, 73, 76, 262–63, 344n11,  
346n6; meaning of term, 28; Moore's,  
326–30, 333, 355n1; of nature,  
140–41, 342n5; power, 84, 86, 268,  
346n8, 347n9; of requisite variety,  
296–97; universal gas, 81
- laws of thermodynamics, 17, 28
- learning: algorithms, 145–69; computer,  
145–50, 168, 348n1; human, xiii, 230–  
55, 318–20, 323; more than copying,  
146; and randomness, 167–69
- left wall effect, 286–88
- Lloyd, Seth, 4, 8, 341n2, 342n9, 346n1
- local minimum, 247
- leukocyte, 182, 349n4
- life: three ways it could have originated,  
306; what is it?, 222–28, 305–14
- light, switchable, 209–10
- lightning, 81, 84, 212
- lion chases: gazelle, 298–9; wildebeest,  
235–36
- logical operation, 46, 48, 51, 162
- logic: circuit, 46, 48; gate, 46, 62, 345n3;  
fallibility of human, 262
- long division, 42–44
- Louvre, 170
- lysozyme, 99–101
- L-Z compression (*see* compression)
- machine (computer) states, 45
- macromolecule, 29, 182, 316, 323. *See*  
*also* antibody; DNA; enzyme; protein;  
RNA
- macrophage, 182, 188
- maize (*see* corn)



- major histocompatibility (MHC) proteins, 188, 349n4
- making sense (brain function), 243
- master engineer, the brain doesn't have one, 244
- Maxwell's theory of electricity and magnetism, 31
- McShea, Daniel, 288, 291, 342n4, 353n8
- meaning, 16, 20–21, 40–41, 44, 67, 193, 343n3
- meiosis, 197
- melanin, 124
- membrane: biological, 98, 109–14, 222–27; potential, 112–14; proteins, 115, 223, 225, 228, 316
- meme, 258, 347n6
- memorization, as opposed to learning, 146
- memories: in brain function, 235–36, 240, 243–44, 250, 252, 273; can't copy human, 146, 242; immortalizing, 336
- memory: cells in immunity, 186; computer, 44–45, 149, 160, 280, 299; fallibility of human, 273; importance for complex adaptive systems, 229
- mental model, 246, 273, 319
- metabolism: cellular, 29, 106, 110, 214, viruses, 105
- metabolism first (origin of life) (*see* biochemistry first)
- micelle, 223
- minimal description, 280
- miracle, 11, 306
- molecular motion, 172, 238. *See also* thermal motion
- molecular tiles, 58–59, 62, 67–68
- molds are templates, 194
- momentum (*see* conservation of momentum)
- money, 265–69
- Moore, Gordon, 327
- Moore's law, 326–30, 333, 355n1
- microstate, 18, 22
- Morse code, 233
- muscle: cells, 110, 178, 232, 240; contraction, 174, 232; control, 240; fibers, 232
- mutation: algorithmic, 152–53, 156–57, 159, 164; in bacteria, 201–2; in living systems, xiii, 29, 38, 124, 128, 138, 140, 196–97, 287, 298; optimum rate, 157; rate, 128, 152, 196–97, 201; role in antibody production, 189, 191, 315 (*see also* hypermutation); sweet spot, 153
- nanotechnology, 326, 330, 332–6
- natural selection, xiii, 6, 29, 38, 138, 145, 200, 202, 204–5, 258, 271
- natural world, underlying orderliness, 169
- negentropy, 22, 277, 284, 343n7, 352n1
- nervous system, 240, 317, 320, 322, 349n2
- network: artificial neural (*see* ANN); biochemical, 212–16, 218, 223, 227–28, 308–12; Boolean, 208; complex adaptive system, 229; concept of, 208; genetic, 139, 216–21; economic, 269; electronic, 209–12; metabolic, 91, 222, 225, 300; neuronal, 233–34, 339, 241, 244–53, 255, 259; protein, 221–22; self-catalytic, 308, 310–14; trading, 267; tripartite supernetwork that characterizes cells, 222, 225, 227–28, 267, 290–91, 311
- neuron: function, 110–14, 245; in brain, 230–55; in brain development, 176–81; as microprocessor, 232, 239; structure of, 111, 180
- neuron to neuron connections, number of in brain, 176–77, 239, 252
- neurons communicating with other neurons, 26, 231–34, 239–44, 250–52, 255, 259
- neuronal branching, 178–81, 316
- neuronal connectivity patterns, 97, 146, 177–80, 193, 233, 239, 242–45, 251–52, 255, 316–18

- neuronal group selection, 177–78
- new kind of science (NKS), 51–58, 66–67, 88, 107, 207, 345n4
- Newton, Isaac, 27–28
- NKS (*see* new kind of science)
- node: of an ANN, 146–48, 244–46; of a graph, 208–9; of a network, 209–22, 228–9, 267, 290–91; of a parse tree, 161–63
- non-localized adaptation, 294, 302, 353n13
- nucleotide: chain, 102–4; monomer of RNA and DNA, 102; pairing, 102, 194; as a unit of information, 143, 258, 313
- numbers, 130–33; big and small (table), 133
- one-max problem, 136, 153–58, 167, 292, 325
- optimization, 292–96
- order (the opposite of disorder), 119–22, 126, 137, 169, 209, 269, 273, 277, 298; for free, 209; perceived, 120
- origin: of animal body plans, 290; of animal brains, 316–8, 337; of complexity, 4; of culture, 320–21, 337; of creativity, 255; denial of evolutionary, 2; of evolutionary algorithms, 321–22, 337; of the immune system, 314–16; of information that specifies a good fit, 145; of instructions, 6, 13, 15, 38, 125–30; of life, 8, 26, 138, 305–16, 322, 336–37, 354n1, 355n4; of purpose, 35; of thought, 318–20, 337
- organ formation, 170–76
- paint-bots, 294–96
- Paley, William, 35, 97–98, 136, 345n17, 347n2
- paradigm shift, 7, 342n7
- paradigms of technology, 327
- parasitic lineages, 275, 291–92
- parse tree, 161–63
- parts: of an automobile, 107, 267, 277–78; of the body, 174, 178, 217, 284; of the brain, 232, 240–43, 257; of cells, 97–98, 222–28; complicated, 207, 209, 279; of economies, 265–69; relationships between, 119–21, 207–9, 225, 277, 280, 284, 298; of salt crystals, 72; of sandpiles, 89; of a snowflake, 171; of systems or of objects, 12, 22, 37, 40, 64, 70, 120, 207–9, 241, 277–79; of a transistor radio, 210–12; of a virus, 105–9; of a watch, 97, 105; of water waves, 82
- paths through possibility space, 123–24, 226, 292, 325
- pattern recognition, 146, 240, 329
- Pauli exclusion principle, 73–74
- penicillin, 201
- personalized medicine, 331
- phenotype, 139–41
- phospholipid, 223, 225, 227
- phospholipid bilayer, 223–24
- phosphorus 32 decay, 123–24
- photoelectric cell, 238
- photosynthesis, 12, 306
- photon, 238
- physical processes, 28, 34, 59, 64, 68–90, 306
- physical structure, computation of, 32, 58–67, 172, 299
- picket fence example, 137
- pilgrims, 272
- pin factory, 267
- pixel, 119–20, 148
- Planck length, 132–33
- plants: breeding, 30, 200, 203–4; development, 138–39; evolution, 29, 285, 326, 352n5
- plasmid, 202
- polymer, 308, 310–14
- polypeptide, 188, 216–7, 311. *See also* protein
- polysaccharide, 181–82, 227, 308, 314, 316

- Popper, Karl, 231, 242, 261, 342n7, 345n18, 350n2, 351n3, 351n8, 352n4
- population: biologists, 258, 271, 342n11; genetics, 25, 142, 248, 351n1
- populations: bacterial, 202; of cells 187; of entities, 24, 116, 128, 144, 153, 294–96; human, 30; of minds, 273; of neurons, 178; of organisms, xiii, 30, 140–43, 258
- possibility space, 123–24, 129, 169, 325
- possibilities, unlimited, 25
- potentially useful structure, 299
- power law, 84–86, 88, 268, 346n8, 347n9
- predicting the next bit, three possibilities, 234
- prediction: in brain function, 246–48; of depth, 301; Moore's law, 326; based on science, 27, 237, 301; statistical, 34
- prespecification, xiii, 115, 209, 229
- probability: appearance of extremely low, 13, 115; distributions used by the brain, 245–46, 251; of existence, 1, 13, 122–25, 135; indistinguishable from zero, 70, 115, 120, 122, 133–35; and instruction use, 13, 25, 93, 115, 130–37; of life, 13; problem, the, 70, 252, 255; of radioactive decay, 123; role in information theory, 18–22, 343n4
- probability distribution, Bayesian brain, 245–46, 251
- problem solving: in science, 261; by the brain, 248, 251–52, 254
- procedures, logical structure of: algorithms, 44; engine of complexity, 128; evolutionary processes, 24; parse trees, 161
- production, in economics, 266
- productivity, 202, 266–67
- programs: complex, 52; computer, 28, 34, 41, 66, 126, 129, 143–44, 171, 192, 281–82; evolutionary, 150; genetic (*see* genetic programs); long, 32, 51, 66–67, 92; one-max, 154; short, 51, 66, 92; structure of, 44; Tartarus, 158
- progress: cultural, 257; evolutionary, 26, 283–86; illusion of, 286; scientific, 28, 342n7; technological, 30, 329, 331
- progressive universe, 283
- promoter, role in gene expression, 219–20
- protein: as cell parts, 97–100, 104, 109, 111–14, 142, 174, 194, 222, 227, 290; conformational change, 109–10; determines phenotype, 139–41, 174, 228; as enzyme (*see* enzyme); fit to something, 98, 105, 107; foreign in body, 181, 189, 191, 314, 316; functions by binding, 98–100, 105, 107, 182, 221–22, 227–28, 258, 315–16, 323; importance of shape, 99, 103–5, 194; membrane, 111–14, 223–25, 316; number of types in organism, 139, 216–17; one-gene-one protein, 102; purpose, 117; regulatory, 139, 218–23, 228, 258, 347n3; role in cellular networks, 214–22; structure, 100–1, 184; synthesis, 99–105, 139, 218–19; virus, 105–10, 139. *See also* polypeptide
- proteome, 221–22, 267, 311
- pump (*see* ion pump)
- purpose: appearance of, 35; concept of, 7; implied need, 116; and instructions, 15, 24, 35–38, 92, 116, 194; of proteins, 117; of screws, 117–18; scientific explanation of, 7, 12, 34–38, 262
- purposefulness, 116
- purposeful: bodies of information, 116, 136, 250; complexity, 6, 34–38, 341n1; detail (in brain), 174, 177; fit, 35–36, 116; information, 15, 23, 37–38, 92, 97, 107, 125, 127, 255, 311, 313; networks, 212; objects, 35, 116, 171; structure, 12, 91–118, 171, 269, 298
- quantum mechanics, xiii, 8, 13, 73–74, 77, 132, 237–38, 346n4, 350n5, 350n6

- radio, 210–12
- radioactive element, 123, 237
- random: assembly of bacteriophage, 107; assembly of objects, 11–12, 70, 135–37; assortment, of chromosomes, 143, 197–99; branching of growing neurons, 179–80; chemical processes, 309–10, 355n5; connections in brain development, 179–80, 191, 316–17; creation of objects, 70, 136–17; generation of character strings, 136, 151, 156–57, 235, 313; generation of DNA sequences, 226; generation of instructions, 115; guessing with selection, 149, 166; mutation with selection, 29, 153, 156, 191, 201; search, 167; trials with selection, 317, 322; variation, 187, 190, 195, 325
- random change: source of creativity, 7, 23, 238, 255, 325; in DNA sequences, 38, 124, 145; information extracted from, 23, 127, 130, 146, 151, 159, 169; hereditary, xiii; in computer programs, 161–63; maximizes future potential, 196; requirement for small, 127, 150, 248, 276; role in antibody production, 184–87, 191; role in the engine of complexity, 127–30, 169, 195–96; role in evolution, 34, 37; with selection, 13, 116, 124, 127, 130, 148, 150, 251, 255, 259–60, 350n2
- randomness: and learning (*see* learning); in liquids and gasses, 238; source of new information, xiv, 23–24, 167
- random number generator in brain, 235
- rate constant, 213
- reaction rate, 215
- recipe, soufflé, 14
- recombination: biological, 143, 197–99, 316; in genetic algorithms, 153, 159, 163, 200, 295, 349n2
- Red Queen, 301, 353n16
- reentry, 243
- regularities in the physical world, 7, 27, 31, 237
- religion, 35, 37, 254, 261, 270–73, 305, 320, 342n11
- religions, evolution of, 270–73
- redundancy, in an object, 21, 50–51, 102, 304, 354n18
- relationships; complicated, 34; simple, 7–8, 32–34
- replicators, 6–7, 309
- representation: in information science, 41–42, 45–46, 48, 51, 150, 161, 193–95, 245; mental, 146, 234, 241–42, 251, 255, 323; as a network, 213
- reproduction: biological, xiii, 98, 140–42, 215; cell, 93, 187, 226; of complex structures, 171, 195; cultural, 259, 263, 266; role in evolutionary systems, 24, 129, 138, 140–43; sexual, 153, 197–98; virus, 105–10, 181
- reproductive success, 38, 142–43, 276
- resources, computational, 280, 296–99, 302
- resting potential, 112–14
- ribosome, 102–4, 218, 222, 227
- RNA: encoding information, 13, 140, 309, 312–13, 323; encoding instructions, 92, 311; enzymatic activity, 309–10, 312, 314, 355n2; messenger, 102–4, 111, 194, 217–20, 229, 347n3; polymerase, 102–3, 218–20, 313; role in protein synthesis, 100, 102–4; self-replicating, 309, 314, 355n2; transfer 103–4, 227, 347n3; world, 309, 312
- Roman numerals, 131, 348n3
- root of a parse tree, 161–62
- rule 110 is universal, 54, 56, 64
- rules: absence of, 122–23, 127–30; of body development, 174, 177–79, 242; complex, 6, 32–34, 52, 66–67, 70, 91–118, 172, 179; of computation, 32, 34, 45–46, 48, 52, 62, 66, 107, 125–27, 324; deterministic, 23, 34, 64, 66, 69; of evolution, 34, 116;

- evolving new, 161–64; physical, 6, 17, 58, 69, 123, 137, 209, 236, 306; selection, 116, 127, 129–30, 204–6, 259, 276, 292, 317, 326; simple, 32–34, 52–58, 67, 69–90, 121, 171–74, 207–8
- sandpile: 83–86, 207, 229, 268; algorithmic, 86–89
- scala naturae*, 283
- scale: importance of size, 132, 278–79, 297; time, 181, 189
- Schrödinger equation, 346n4
- science: central activity, 26; evolutionary nature of, 261–65; nature of, 26, 250, 301; predictions based on, 27; technology based on, 30–31
- scientific: advancement, 261, 342n7; creationism, 262; notation, 19, 132–34; understanding, 2, 8, 26–31, 36, 132, 138, 261, 305, 323
- screwdriver, example of, 36, 92–97, 115, 117, 171, 174
- sculpture, 170–71
- searching large spaces, 156, 167–68, 263–64, 281–82
- seed sequence or row, 59–61, 63–64, 66
- selection: of ANN weights, 149, 244; artificial, 30, 200–4; in brain evolution, 316–18; as constraint, 137, 273; copying and, 126, 189–91; criteria or rules, 24, 116, 125, 127, 129–30, 205–6, 259, 276, 292–93; cumulative, xiii, 37, 116, 244, 255, 308; and the engine of complexity, 116, 127–30, 142, 169, 176, 196, 204–5, 252–55, 259–60, 273, 308, 312, 322, 324; in evolution, 6, 24, 35, 138; in evolutionary algorithms, 152, 164, 322; favoring depth, 300, 326; importance of consistent criteria, 37, 124, 151, 204, 273; is non-random, 116, 124, 130, 137; natural, xiii, 6, 29, 36, 38, 127, 129, 138, 142, 145, 205, 258, 276, 293, 341n4; pillar of Darwinism, 6; power of mild, 142; repeated, 7, 30, 36, 38, 116, 125–26, 137, 181, 254–55; role in antibody production, 186–89, 190–91; role in body development, 173–6, 189–90, 315–6; role in brain development, 177–81, 191; role in economies, 265; role in the origin of life, 305–14; role in religion, 271–72; role in science, 263–64; during sculpting, 172; in thought processes, 319–20, 345n18, 349n2, 350n2, 351n9
- selective remembering, 259
- self-assembly: in living organisms, 62, 222; molecular tiles, 58–64, 345n7; protein structures, 97–99; puzzle, 64–65; viruses, 105–9
- self-awareness, 231
- self-organization, 80, 85, 88, 115, 229, 313, 347n9
- self-organized criticality, 84–86
- self-replication, 309, 312–16
- sensory input to the brain, 231, 239–40, 243, 245–47, 251–52, 255, 318–19, 323
- Shannon, Claude, 18, 20
- Shannon equation, 343n4
- Shannon information (*see* information)
- Sierpinski gasket, 54–55, 61–63
- simple rules in computation (*see* rules, simple)
- slow growth (property of deep objects), 299, 327
- small improvements are not improbable, 24
- Smith, Adam, 267
- Smith, Joseph, 272
- snowflake, 32, 34, 68–70, 92, 97, 171–73
- Sobey, Dr. Ed, 249, 253, 351n12
- social evolution (*see* evolution, social)
- sodium chloride, 72–74
- solar system, as an equilibrium system, 71, 89

- solar system, formation of, 75–78
- sound waves, theory of, 27
- soufflé, recipe for, 14, 343n1
- space: of all possibilities, 122–23; of all possible solutions, 164, 231; as a resource, 280, 297, 299
- spaces, solution, 164–69, 293
- species extinction, 84, 129, 196, 352n5, 354n16
- Spencer, Herbert, 257
- surprise, Bayesian brain theory, 246–48, 253
- speed of light, 133
- splicing, alternative, 217
- stable states, far from equilibrium, 71, 208, 214, 268, 291
- state: of a cellular automaton, 52; of a computer, 42, 45–46, 280; critical, 84–85, 88–89; equilibrium, 71, 74; initial, 97; lowest energy, 71; mental, 235, 239–43, 245, 250; of a system, 18, 21, 41–42, 67, 88, 343n4
- states: contingent on past states, 123–24; of ion channels, 112; statistical inference (in brain), 246, 251
- stem cells, 185, 188
- steps: of an algorithm, 151, 153, 158; of a computation, 46, 52, 55–7, 299; of evolution, 29, 315, 319; of long division, 42–44; specified by instructions, 13–15, 37, 93–96, 99
- structure: dynamic, 71; equilibrium, 71, 75, 78; hierarchy, 172–73; independent of parts, 81–83; nonequilibrium, 71, 78–82, 89, 110; purposeful (*see* purposeful structure); static, 71
- structure for free, 68, 72, 74, 88–89, 92–93, 95, 99, 104, 116, 346n3
- structure formation, cumulative, 171
- symmetries, 278–79
- synapse: 114, 178–80, 231, 233, 239, 244–45, 247, 316; adjustable, 233, 245, 248; inhibitory, 232–33
- system: meaning in physics, 17, 18; isolated, 17
- $T_4$  (*see* bacteriophage)
- Tartarus, 158–60
- T-cell, 182–83, 186–89; receptor, 182–83, 188, 315–16
- taxonomic families, 282–83, 352n5, 353n16
- technology: merging with, 334; role of instructions, 37, 66–67, 255; role in science, 30–31; three phases, 327
- telephone, game of, 259
- television screen image, 119
- Teller, Astro, 158, 348n3
- template: role in copying, 194; DNA strand, 194, 226; row (*see* seed row)
- texting, cell phone, 145
- three body problem, 237
- theory: computer (*see* computer theory); of evolution, 8, 28–30, 138, 143–44; general theory of evolution, xiv, 143; information (*see* information theory); of everything, 69; of sound waves, 27; theory vs. law, 28
- thermodynamics, 17, 20, 71, 264, 284, 298; of evolution, 2, 5; first law of, 17, 28; second law of, 17
- thinking with hands, 249
- thought: brain mechanism, 232–34, 251; as computation, 324, 336; deterministic, 238, 337; information in, 232–33, 236, 250, 255; is the result of evolutionary computation, 192, 250–54, 323, 350n2
- thinking, origin, 318–20
- “thinking up” is not an explanation, xiii, 136
- thunderstorm, example, 80–82, 89, 115–6, 207
- time: as a resource, 280–82, 297–301; scale, 181, 189, 278, 318
- Toyota Motor Corporation, 266
- trading, 265–67

- training (of ANN), 148–49, 244–45  
 trajectories through possibility space, 325  
 transcription factor, 219–22  
 transistors, number, 326–27  
 translation (protein synthesis), 102–5  
 transposase, 315  
 transposon, 315  
 tripartite supernetwork (*see* network, tripartite)  
 triplet code (*see* code triplet) trust,  
     importance of for cooperation, 271  
 truths: mathematical, 28, 34; scientific, 6;  
     ultimate, 263  
 Turing, Alan, 42, 46  
 Turing Machine, 44, 62  
 TV interview, 334–35  
 Type I and Type II objects, 15, 25, 34–37,  
     70  
 uncertainty: role in brain function, 239,  
     246–47, 351n10; and chaos, 237;  
     with early earth theories, 306; and  
     entropy, 18; and information theory,  
     20; principle, 132; and quantum  
     mechanics, 238  
 understanding (*see* scientific  
     understanding)  
 United Church of Christ, 272  
 universe: age, 133; is based on  
     information, 8; as a computation,  
     xiv, 4, 32, 58, 341n2; cannot be  
     deterministic, 8, 238; the interplay of  
     four things, 236  
 universal gas law, 81  
 unknowable, 168, 237–38  
 uranium 235 decay, 123  
 U.S. government debt, 131, 133  
 value, in economics, 265–66, 269  
 Van Valen, Leigh, 301, 353n16  
 variation: role in algorithmic evolutionary  
     processes, 150–53; role in body  
     development, 176, 190, 192, 318,  
     349n2; role in cultural evolution,  
     264–65; role in engine of complexity,  
     127–28, 325; role in immune system,  
     187, 316; role in natural evolutionary  
     processes, 35, 138, 140–43; role  
     in non-natural selection, 195–200,  
     204–5, 276, 294  
 Venter, Craig, 330  
 vertebrae, variability of, 291  
 vesicle, 217, 224  
 virus: use in biotechnology, 205, 331–32;  
     drug resistance, 202; evolution of, 293;  
     infection, 106, 110, 114, 181, 189,  
     293; mutation rate, 197; self-assembly,  
     105–9. *See also* bacteriophage  
*Watchmaker, The Blind*, xii, 4, 35, 98,  
     341n3, 344n13, 345n17, 347n2  
 watchmaker analogy, 97–99, 105  
 water: cohesion, 82; waves, 82–83, 88–89,  
     115–16  
 Watson, James, DNA sequence, 330  
 waves: of activity (action potential),  
     110–14; ocean or lake, 71, 81–84;  
     sound, 27  
 wealth generation, 269  
 weights: ANN, 146–49; brain theories,  
     244–47  
 Wheeler, John, 8  
 Wilson, David Sloan, 270–72, 342n11  
 Winfree, Erik, 59, 61, 345n6  
 Winged Victory of Samothrace, 170  
 Wolfram, Stephan, 52, 345n4  
 world model (in the brain), 240, 245–47,  
     319  
 writing (a form of language), 256–57  
 Wright, Sewall, 164, 349n5  
 XOR logic function, 46–48, 62–64  
 ZIP file, 51, 302





